Training-efficient Deep Reinforcement Learning for Safe Autonomous Driving

Zhen Tian

Supervisor: Dr. Dezong Zhao

SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

SCHOOL OF ENGINEERING

COLLEGE OF SCIENCE & ENGINEERING

11/12/2024

To my supervisor and life-mentor,

***Dr. Dezong Zhao**,*

my parents,

***Mr. Xiaokai Tian and Ms. Jinhong Bai**,*

my pal,

***Mr. Zhihao Lin**,*

my outer-supervisor,

***Dr. Wenjing Zhao**,*

my academic-guiders,
***Dr. Chongfeng Wei and Prof. David Flynn**, whose guidance, support, and encouragement made this work possible.*

Abstract

Autonomous driving holds the potential to transform the transportation industry, offering significant improvements in safety, efficiency, and convenience. However, traditional model-based planning approaches struggle to address the complexities and uncertainties of real-world driving environments. This thesis employs deep reinforcement learning (DRL) to achieve safe and efficient autonomous driving using realistic simulation settings and evaluation based on rational criteria.

The proposed framework integrates five key factors—driving safety, driving efficiency, training efficiency, unselfishness, and interpretability (DDTUI) to ensure reliable and optimal decision-making across various driving scenarios. The research addresses two primary applications: highway driving and autonomous racing. In highway driving, the DRL-based framework demonstrates superior performance compared to popular baseline algorithms, improving safety and efficiency. In autonomous racing, an extreme case of autonomous driving, the framework is adapted to manage high velocities and safe control, achieving fewer collisions, faster lap times, and reduced training time in comparison to benchmark algorithms.

This thesis contributes to the field by advancing RL-based planning techniques and establishing a design methodology for integrating key factors in autonomous driving. The results of this study provide evidences of the development of safer, more efficient, and interpretable autonomous driving systems. Finally, key achievements are summarized, limitations are discussed, and future research directions are proposed.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to express my deepest appreciation to all those who have helped me complete this dissertation. First, I give all glory and thanks to God for granting me the courage, strength, and blessings needed to accomplish this work.

I am sincerely grateful to my supervisor and life mentor, Dr. Dezong Zhao, for his unwavering guidance, encouragement, and support throughout my Ph.D. journey. Dr. Zhao was always there to help me overcome academic challenges, often organizing face-to-face meetings to provide clarity and direction. His generosity in sharing valuable learning materials, relevant research papers, writing skills, and invaluable life experiences has been instrumental to my growth. Dr. Zhao dedicated additional time to help me develop my academic capabilities, especially given my initial challenges. His patience and kind mentorship enabled me to navigate each academic hurdle, building my knowledge from a foundational level. Without his guidance, I could not have achieved these milestones.

I also express my deepest gratitude to my parents, Mr. Xiaokai Tian and Mrs. Jinhong Bai, for their unwavering encouragement throughout this long journey. Their constant support, love, and unconditional patience have been my greatest source of strength and motivation.

I am immensely thankful to my friend, Mr. Zhihao Lin, for his invaluable support and care throughout my Ph.D. journey and daily life. His academic and life experience helped me quickly improve my understanding and adapt my routines. Thank you, Zhihao, for being the best friend I could have during this journey.

I am also very grateful to my external supervisor, Dr. Wenjing Zhao, for her dedication to helping me refine my academic skills. Her careful review of my work, face-to-face discussions, and emphasis on key areas for improvement significantly contributed to my development. Her encouragement during difficult times boosted my confidence and belief in positive outcomes.

I extend my thanks to Dr. Chongfeng Wei and Prof. David Flynn for their valuable advice and insights throughout this journey. I am also thankful to my research team, including Mr. Zhaoan Ye, Mr. Yansong Jiang, Mr. Xiao Yi, Mr. Qikun Chen, and all my colleagues, for their support and for alleviating the pressures of this Ph.D. journey. Thank you for the warm greetings, continuous support, and valuable discussions that made this journey smoother.

Declaration

I declare that, except where explicit reference is made to the contribution of others, that this dissertation is the result of my own work and has not been submitted for any other degree at the University of Glasgow or any other institution. This thesis has been written and compiled by the author, Tian Zhen, and certifies that the thesis presented here for examination for the PhD degree at the University of Glasgow. Part of this research work are published and are detailed below.

- **Zhen Tian**, Dezong Zhao, Zhihao Lin, Wenjing Zhao, David Flynn, Yuande Jiang, Daxin Tian, Yuanjian Zhang, and Yao Sun. "Efficient and Balanced Exploration-driven Decision Making for Autonomous Racing Using Local Information." *IEEE Transactions on Intelligent Vehicles* (2024). (doi: https://eprints.gla.ac.uk/330387/2/330387.pdf)

- **Zhen Tian**, Dezong Zhao, Zhihao Lin, David Flynn, Wenjing Zhao, and Daxin Tian. "Balanced Reward-Inspired Reinforcement Learning for Autonomous Vehicle Racing." *In 6th Annual Learning for Dynamics and Control Conference*, pp. 628-640. PMLR, 2024. (doi: https://proceedings.mlr.press/v242/tian24a/tian24a.pdf)

- **Zhen Tian**, Dezong Zhao, Zhihao Lin, Wenjing Zhao, David Flynn, Daxin Tian, and Yao Sun. "Balanced Exploration and Attention-Inspired Decision Making for Autonomous Driving." *IEEE Transactions on Vehicular Technology* (2024). (Under Review)

- **Zhen Tian**, Zhihao Lin, Dezong Zhao, Wenjing Zhao, David Flynn, and Chongfeng Wei. "Evaluating Scenario-based Decision-making for Interactive Autonomous Driving Using Criteria Matrix: A Survey." *IEEE Transactions on Intelligent Transportation System* (2024). (Under Review)

- **Zhen Tian**, Zhihao Lin, Dezong Zhao, Wenjing Zhao, and David Flynn. "SPL-SLAM: Point-line SLAM System Enhanced by Dynamic Object Segmentation and Removal." *IEEE Transactions on Automation Science and Engineering* (2024). (Under Review)

- **Zhen Tian**, Zhihao Lin, Dezong Zhao, Wenjing Zhao, David Flynn, Xiaoxiang Na, and Chongfeng Wei. "Cooperative and Collision-free Planner for Automated Driving on Multi-lane Ramps with Static Block and Dynamic Competitors." *IEEE Transactions on Automation Science and Engineering* (2024). (Under Review)

- **Zhen Tian**, Zhihao Lin, Dezong Zhao, Wenjing Zhao, David Flynn, Xiaoxiang Na, and Chongfeng Wei. "A Risk-aware Spatial-temporal Trajectory Planning Framework for Autonomous Vehicles Using QP-MPC and Dynamic Hazard Fields." *IEEE Transactions on Vehicular Technology* (2024). (Under Review)

- **Zhen Tian**, Zhihao Lin, Dezong Zhao, Wenjing Zhao, David Flynn, Xiaoxiang Na, and Chongfeng Wei. "Adaptive Evolutionary Framework for Safe, Efficient, and Cooperative Autonomous Vehicle Interactions." *IEEE Transactions on Intelligent Transportation System* (2024). (Under Review)

- Zhihao Lin, **Zhen Tian**, Qi Zhang, Hanyang Zhuang, and Jianglin Lan. "Enhanced Visual SLAM for Collision-Free Driving with Lightweight Autonomous Cars." Sensors 24, no. 19 (2024): 6258. (Equal Contribution with Zhihao)

- Zhihao Lin, Qi Zhang, **Zhen Tian**, Peizhuo Yu and Jianglin Lan, "DPL-SLAM: Enhancing Dynamic Point-Line SLAM Through Dense Semantic Methods," in IEEE Sensors Journal, vol. 24, no. 9, pp. 14596-14607, 1 May1, 2024, doi: 10.1109/JSEN.2024.3373892.

- Zhihao Lin, Qi Zhang, **Zhen Tian**, Peizhuo Yu, Ziyang Ye, Hanyang Zhuang and Jianglin Lan. "SLAM2: Simultaneous Localization and Multimode Mapping for indoor dynamic environments." Pattern Recognition 158 (2025): 111054.

Tian Zhen

Chapter 1

Introduction

1.1 Motivation

Autonomous driving has emerged as one of the most transformative innovations of the 21st century, poised to revolutionize the transportation industry by offering the potential for enhanced safety, efficiency, and convenience in daily life. By eliminating human error, which accounts for a significant proportion of traffic accidents, autonomous vehicles (AVs) are expected to drastically reduce accident rates, decrease traffic congestion, and optimize fuel consumption. Despite this potential, the path to fully autonomous vehicles faces considerable challenges in terms of safety, reliability, and adaptability to complex, real-world environments.

Historically, model-based planning methods have dominated the decision-making field of autonomous driving. These methods rely on accurately predefined models of both vehicle dynamics and the driving environment, utilizing optimization techniques to compute trajectories and ensure safe navigation. Such approaches work well in structured environments, where the traffic dynamics and road conditions are relatively predictable. However, in unstructured or highly dynamic environments, model-based planning reveals several key shortcomings. These include sensitivity to model inaccuracies, difficulty in real-time adaptation to unknown scenarios, and limited robustness in handling uncertainties and variations in the driving environment. As the complexity of driving scenarios increases, the computational burden associated with model-based methods becomes unsustainable, limiting their practical application in real-time decision-making. Additionally, these methods struggle to cope with the unpredictable behavior of human drivers and other road users, leading to suboptimal performance in mixed-traffic conditions.

To address these limitations, learning-based planning methods, particularly those involving reinforcement learning (RL), have gained significant attention in recent years. Learning-based approaches shift the paradigm by enabling autonomous vehicles to learn driving policies directly from data, rather than relying on pre-built models. These methods can capture complex behaviors and interactions with other road users, adapting to diverse driving environments by learning from experience. Reinforcement learning, in particular, allows the vehicle to learn optimal policies through trial-and-error interactions with its environment, improving its ability to handle uncertainties and respond to novel situations. Additionally, imitation learning (IL) can be employed to mimic expert human drivers, further enhancing the driving policy's performance by leveraging demonstrations.

While learning-based methods present substantial advancements, they also introduce new challenges, in autonomous driving. A critical aspect of designing robust learning-based autonomous driving systems is the identification of appropriate factors that influence the vehicle's decision-making process. These factors should encompass essential aspects such as driving safety, operational efficiency, vehicle control, cooperation with other vehicles, and interpretability of the learned policies. Each of these factors plays a pivotal role in ensuring that the autonomous vehicle behaves safely and reliably under a wide range of driving conditions.

First and foremost, safety is the highest priority in autonomous driving. It is essential to ensure that learning-based methods incorporate safety constraints directly into the training process to avoid undesirable behaviors, such as collisions or near-miss scenarios. Efficiency, in terms of both fuel consumption and travel time, is another critical factor, particularly when considering the deployment of autonomous vehicles at scale. In addition to safety and efficiency, control over the vehicle's actions, including maintaining stability and responding to environmental changes, is crucial. Furthermore, cooperative behavior has become increasingly important in recent times. For instance, an AV and a HDV may select a combination of actions that maximizes their total combined profit. While this outcome might not represent the maximum profit for either individual party, it achieves the highest total profit overall and can be regarded as cooperative behavior. Cooperative driving is particularly crucial as AVs must coexist with human drivers and

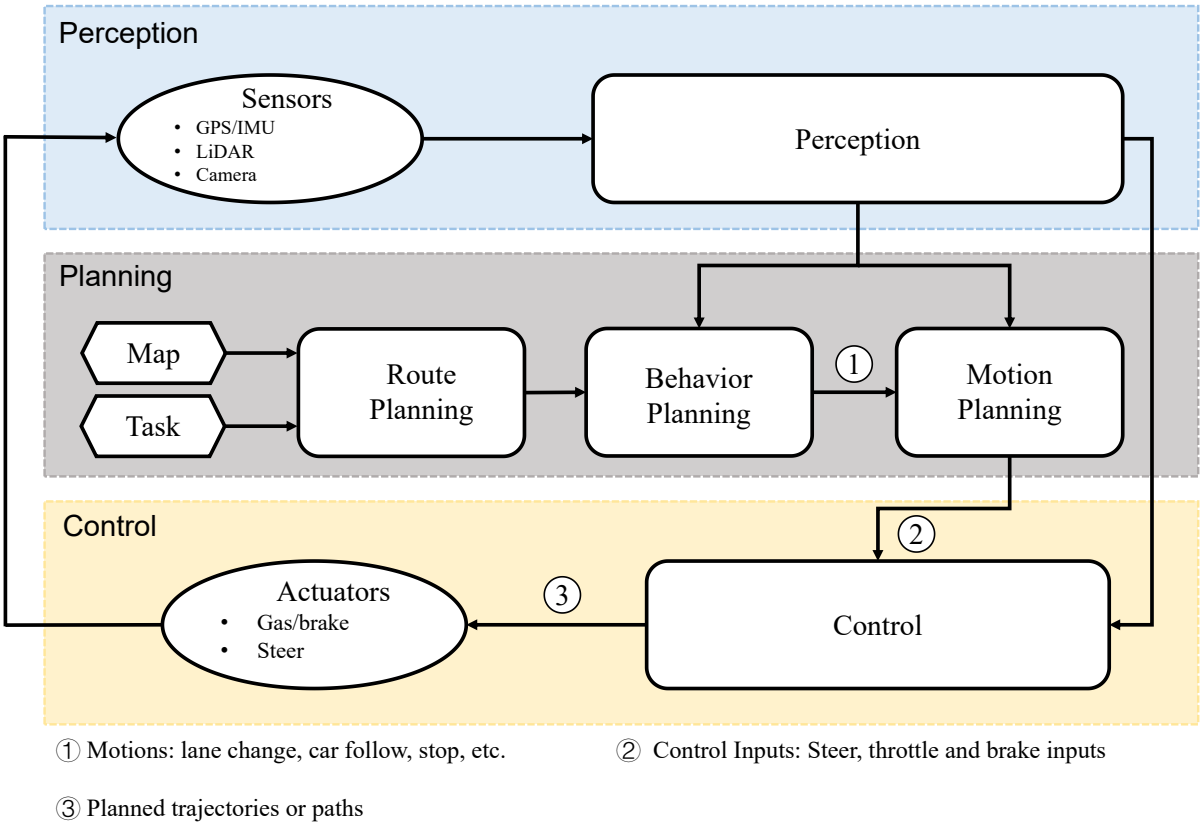


Figure 1.1: The framework of autonomous driving.

other AVs in mixed-traffic environments. Autonomous driving systems must be able to engage in cooperative maneuvers, such as lane changes, merging, and overtaking, in a way that promotes smooth traffic flow and minimizes disruptions. Lastly, interpretability, or the ability to understand and explain the decisions made by the vehicle, is becoming a necessity, particularly for regulatory compliance and user trust in AVs. Fig. 1.1 illustrates the framework of autonomous driving. The autonomous driving framework consists of three main modules: perception, planning, and control. The perception module uses sensors such as GPS/IMU, LiDAR, and cameras to gather data, which is processed to understand the surrounding environment. The planning module has three sub-components: route Planning, behavior planning, and motion Planning. Route planning determines the high-level path based on map and task inputs, while behavior planning decides actions like lane changes and car-following maneuvers. Motion planning then generates specific trajectories to follow. Finally, the control module takes the planned trajectories and com-

putes the necessary actuator commands (e.g., steering, throttle, brake) to execute them. This integrated approach ensures the autonomous vehicle can perceive its surroundings, plan an optimal route, and control its movement to reach the intended destination safely and efficiently.

To design a learning-based planning method that can adequately address these factors, it is necessary to impose proper rules and constraints during both the training and execution phases. The learned policies must adhere to driving norms, regulations, and safety protocols, while simultaneously ensuring optimal performance in terms of efficiency and control. This requires careful design of the reward functions in reinforcement learning, where the reward signal must reflect the trade-offs between safety, efficiency, and other key factors. For example, collision avoidance penalties, lane-keeping rewards, and cooperative behavior incentives must be integrated into a unified framework that promotes desirable driving behavior.

Moreover, planning methods must be adaptable to different driving contexts, ranging from highways to urban environments, where the challenges vary significantly. In highway driving, the focus may be on efficiency and cooperation during high-speed lane changes, while urban driving emphasizes low-speed maneuvering, pedestrian safety, and adherence to traffic rules. Thus, the ability to design flexible and context-aware learning-based methods is another important consideration.

In light of these challenges, this thesis is motivated by the need to explore learning-based planning methods that can systematically integrate multiple critical factors. The goal is to develop a comprehensive framework that ensures safe, efficient, and interpretable decision-making for AVs. By leveraging advanced reinforcement learning techniques and carefully designed reward functions, this thesis aims to advance the field of autonomous driving by developing a rationale evaluation framework for DRL-based decision-making, incorporating five key evaluation factors: driving safety, driving efficiency, training efficiency, unselfishness, and interpretability (DDTUI). This thesis further aims to design and validate the DRL algorithm to address current DRL shortcomings by enhancing safety, efficiency, and interpretability while achieving faster convergence and lower collision rates in complex highway scenarios. Additionally, the research aims to improve DRL adaptability

in extreme driving conditions through curiosity-driven exploration and balanced reward structures, leading to superior lap times, stability, and safety in autonomous racing. Both DRL algorithms for highway scenario and extreme conditions are designed to address all five DDTUI factors simultaneously, promoting well-rounded decision-making strategies adaptable to real-world autonomous driving challenges. Through this, we seek to overcome the limitations of traditional model-based approaches and push the boundaries of what is possible with autonomous vehicle technologies.

1.2 Current Challenges in Autonomous Driving

Autonomous vehicles (AVs) face significant challenges in making reliable decisions when interacting with human-driven vehicles (HDVs). This challenge is primarily due to the difficulty of accurately predicting the intentions of HDVs. Road traffic crashes cause significant fatalities and serious injuries, reflecting the global issue of millions of lives lost annually [1]. Since 2021, over 900 Tesla crashes involving driver-assistance systems have been reported [2]. Despite unresolved safety issues, the number of AVs is projected to surpass 50 million by 2024 [3]. These statistics underscore the critical need for improving safety in autonomous driving. With a safe decision-making system, AVs have the potential to significantly decrease road crashes caused by human errors such as fatigue, distraction, and delayed reactions [4]. Moreover, AVs are capable of making optimal decisions faster than human drivers, thereby enhancing traffic efficiency [5].

There are several typical driving scenarios, such as highways, roundabouts, on-ramping merging, and unsignalized intersections, each characterized by distinct road features and scenario-specific requirements. Autonomous driving in such scenarios is depicted in Fig. 1.2. For example, on-ramp merging involves completing lane changes well in advance of any obstructed roadway, while navigating a roundabout requires seamlessly exiting at the intended point. Achieving these scenario-based requirements relies heavily on precise and timely operational decision-making in real time. Operational decision support for AV driving includes perception, planning, and control modules. The perception module consists of onboard sensors that continuously perceive the surrounding environment. The perceived data is processed through perception algorithms, such as YOLO methods [6, 7].

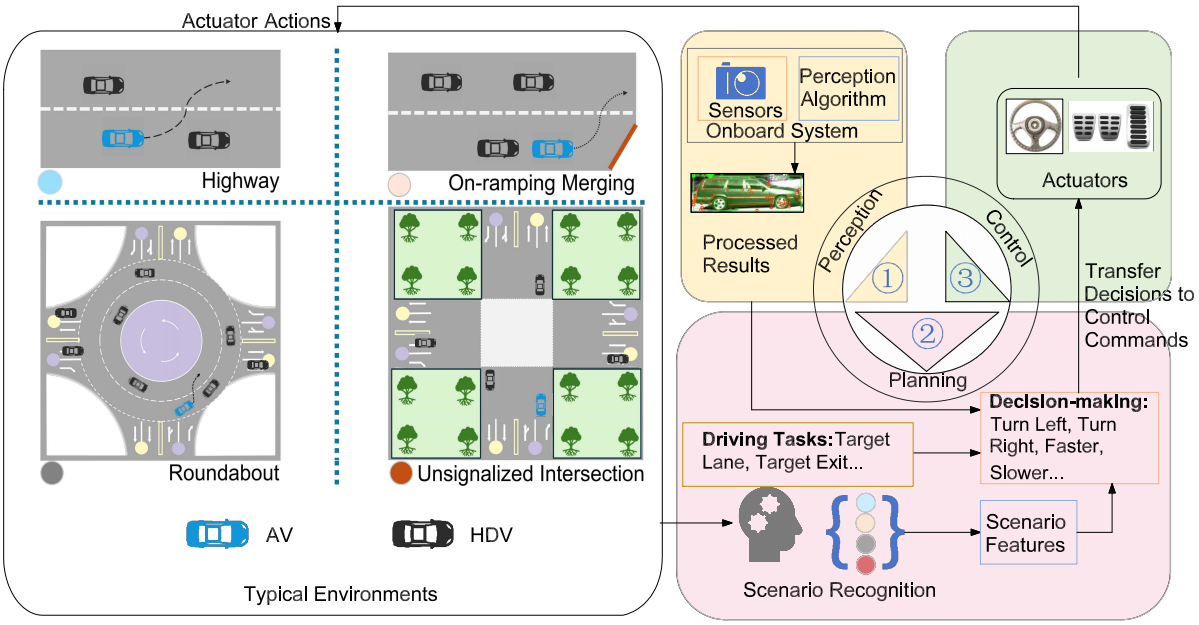


Figure 1.2: Autonomous driving in different scenarios.

The planning module handles driving tasks based on scenario recognition. Subsequently, the motion planner generates discrete decisions and converts them into feasible trajectories. These feasible trajectories are then transmitted to the control module to generate control commands, which are sent to the vehicle's actuators. The actuators, including the steering wheel and pedals, receive and execute the control commands to drive the vehicle.

The interactions between AVs and HDVs are complex and therefore continuous decision-making is required, such as lane changes or braking [8]. The model-based, simple guidance, and learning-based methods are commonly used in interactive driving with HDVs.

1.3 Shortcomings of Model-based Autonomous Driving

There are mainly four types of model-based approaches. The first model-based approach aims to predict the intentions or trajectories of HDVs, but heavily relies on rule-based classification. For example, [9] predicts the trajectories of HDVs within a fixed time window. However, the time required for a lane-changing maneuver may exceed this fixed time window. The second model-based approach is to make decisions using robust control methods, such as the min-max model predictive control [10], as illustrated in Fig.

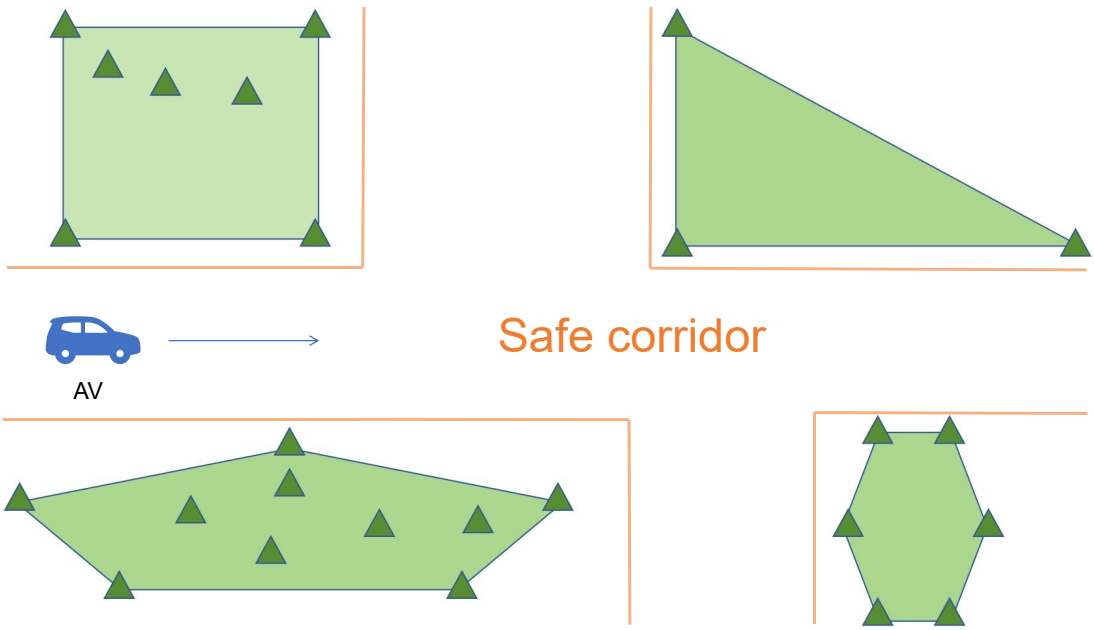


Figure 1.3: This figure highlights a critical approach for ensuring safe autonomous driving in the presence of uncertainty. In autonomous driving, uncertainties arise from various factors, such as sensor noise, environmental variability, or unpredictable behavior of other road users. By predicting the system's states as sets rather than single points, the approach accounts for these uncertainties, providing a robust foundation for decision-making. The green-shaded convex hulls represent the range of possible future states, ensuring that the true state of the vehicle is always contained within these regions. The safe corridor enclosed by orange marigins is crucial for safe planning and control, as it allows the autonomous vehicle to execute actions that maintain safety margin and avoid collisions.

1.3. However, robust control methods make excessively cautious decisions based on a worst-case scenario assumption [11]. These methods are not suitable for most real traffic environments because worst-case scenarios are rare in real-world settings. Furthermore, decisions made for worst-case scenarios negatively impact driving quality, such as resulting in slower driving speeds.

On the other hand, the game theory, the third model-based approach, has gained popularity recently. Game theory includes cooperative and non-cooperative games, both relying on equilibrium models. However, these models fail to capture the complexities of real-world driving, which are characterized by uncertainties and do not adhere to a regular equilibrium framework. In the real world, drivers often exhibit a wide range of behaviors that deviate from purely rational actions. Moreover, the game-based decision-making normally divides the driving styles into three categories: conservative/cautious, moderate/normal, and aggressive. As illustrated in Fig. 1.4, the driving performance of the host vehicle (HV) interacting with V2 is presented under three different driving styles [12].

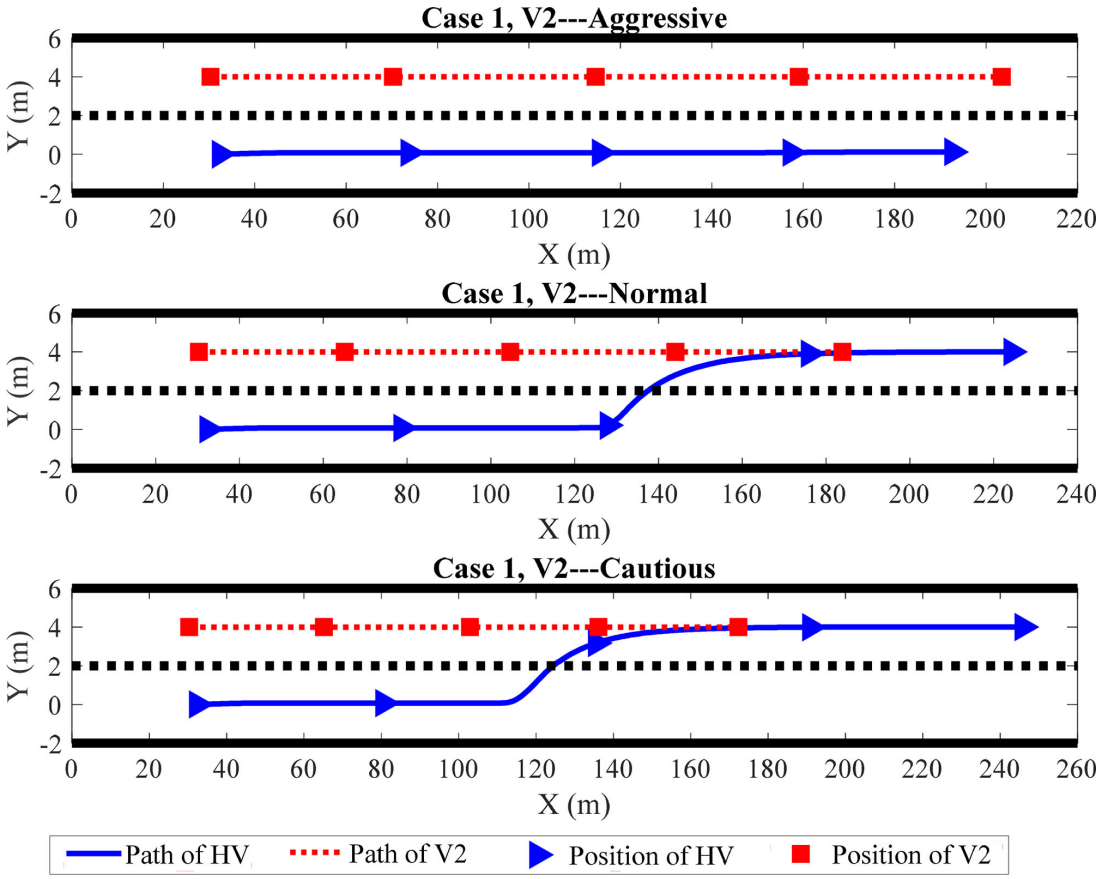


Figure 1.4: Results of the decision making and path planning considering different driving styles of obstacle vehicles [12].

However, real-world driving includes a wide range of driving patterns, which are much more complex and difficult to model. Therefore, game-theoretic approaches may struggle to handle interactions with HDVs that do not behave as expected, potentially leading to unsafe or suboptimal decisions.

Therefore, model-based methods are unable to handle interactive driving with HDVs effectively. Additionally, the fourth model-based approach, including collision-avoidance methods [13] and Voronoi diagram-based methods [14], is unable to safely respond to movable objects. Real-world collisions between HDVs and vehicles equipped with advanced driving system (ADS) assistance are summarized in [15]. As illustrated in Fig. 1.5, 79 % of accidents involve HDVs hitting AVs, and 21 % of that involve AVs hitting HDVs. Therefore, achieving collision-free interactions with HDVs are still to be addressed. [16], which applied APF to guide AVs in lane changes while maintaining a safe distance from surrounding HDVs.

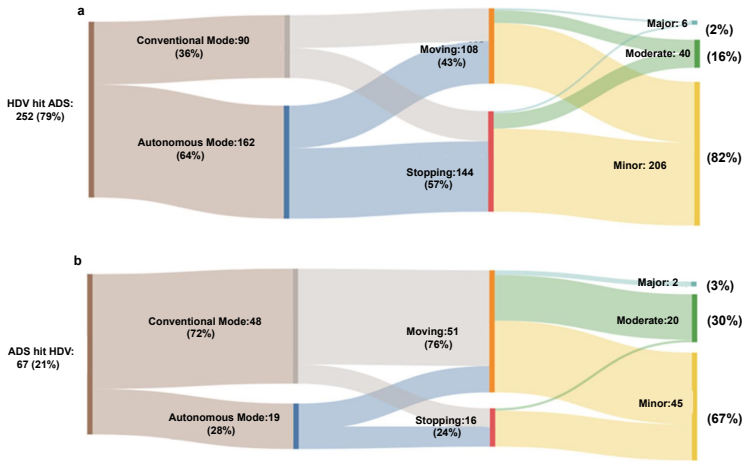


Figure 1.5: Rear-end accident conditions between ADS and HDV: (a) Rear-end accidents that HDV hit an ADS from behind with a sample of 252; (b) Rear-end accidents that ADS hit an HDV from behind with a sample of 67 [15].

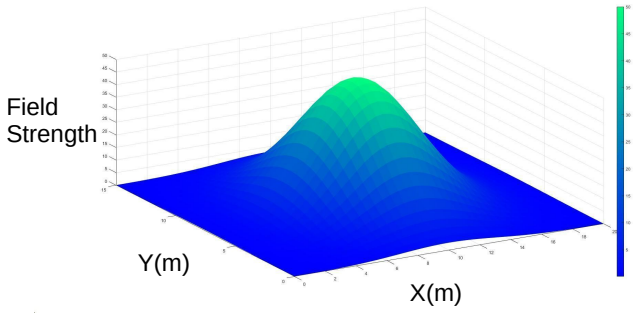


Figure 1.6: APF.

Compared to the aforementioned methods, simple guidance methods, such as risk-quantified fields, are widely used because they do not need to predict HDVs' intentions or make excessively cautious decisions [17]. The artificial potential field (APF) is a typical example, which can guide the AV to the target lane without collisions by utilizing attractive and repulsive force fields [16]. However, APF assumes that all areas around the vehicle have the same level of risk because it calculates risks toward the central point. This assumption differs from reality, where the front of a car faces more danger than other parts. Additionally, APF is difficult to generalize across different scenarios without prior knowledge of the entire environment [18]. Fig. 1.6 suggests an example APF, where the green-shaped areas are with higher potential field compared to blue shaded-area.

1.4 Advantages of RL-based Autonomous Driving

To promote collision-free interactions, a large number of interactions are needed to exclude risky actions, taking into account the uncertainties in decision-making and the varying driving conditions of HDVs. Learning-based methods facilitate the exploration of control strategies by allowing full interaction with the mixed-traffic environment. These methods enable AVs to learn and adapt to complex driving scenarios through iterative interactions and feedback. Machine learning (ML) [19, 20] focuses on developing algorithms to make decisions based on data, including supervised, unsupervised, and reinforcement learning. Supervised learning trains models on labeled data, supporting tasks like classification [21,22]. However, supervised learning is less suited for implementation in real driving environments, as labeling complex driving scenarios exhaustively is challenging and impractical. Unsupervised learning methods are particularly suitable for interactive driving as they do not require labeled data, allowing agents to learn decision-making strategies independently. Unsupervised machine learning has demonstrated robust performance across a range of driving scenarios [23]. However, unsupervised learning often struggles with generalization in highly dynamic environments. Reinforcement learning (RL) is a powerful technique for making optimal decisions in dynamic environments [24, 25]. RL involves an agent that interacts with its environment and learns safe control strategies through a reward-based framework. The adaptability of RL makes it ideal for interactive driving, where the environment is constantly changing, and the AV must adjust its behavior accordingly.

RL offers significant advantages over model-based, supervised, and unsupervised learning approaches for autonomous driving. Model-based methods, such as trajectory prediction and robust control, often struggle in complex, dynamic environments due to their reliance on fixed models and worst-case assumptions, resulting in overly conservative decisions and suboptimal driving quality. Game-theoretic approaches, while promising, fail to capture the diversity of real-world driving behaviors, leading to challenges when interacting with human-driven vehicles (HDVs). Collision-avoidance and APF-based methods further suffer from static risk assessments and limited generalizability across varying scenarios.

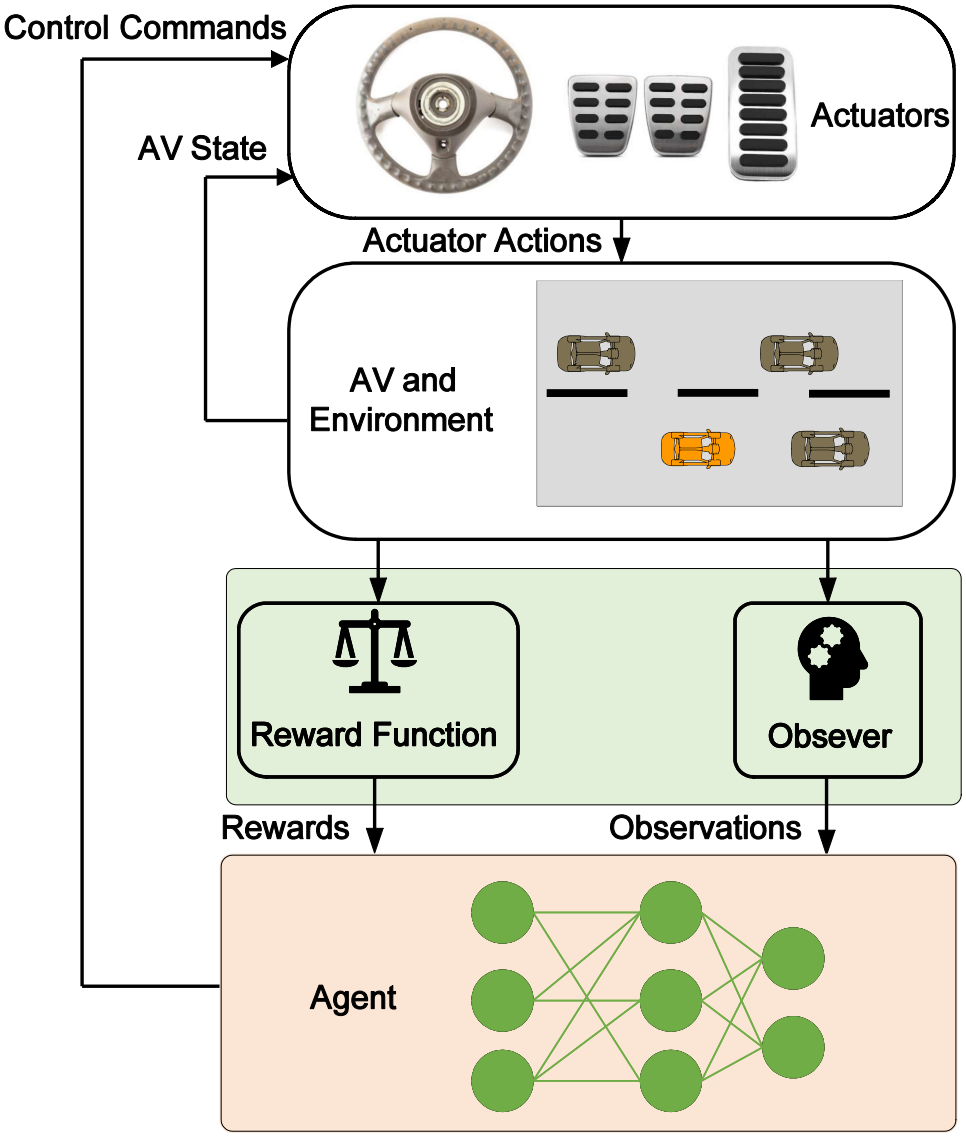


Figure 1.7: DRL-based autonomous driving system

In contrast, RL enables adaptive decision-making by continuously interacting with the environment, allowing autonomous vehicles to respond effectively to unpredictable HDV behaviors and changing traffic conditions. Unlike supervised learning, which relies on extensive labeled datasets and predefined scenarios, RL can explore new situations and adjust its strategies dynamically. Moreover, while unsupervised learning focuses on pattern recognition without task-specific optimization, RL directly optimizes decision-making based on defined objectives such as safety, efficiency, and driving comfort. These advantages make RL a superior choice for autonomous driving, ensuring robust, adaptable, and context-aware decision-making in complex and dynamic environments.

Deep reinforcement learning (DRL) is an advanced form of RL that combines the principles of deep learning [26, 27] with RL. By utilizing deep neural networks to approximate complex value functions, DRL enables agents to learn directly from perceptual inputs, such as sensory data. This capability allows DRL to handle more complex and real-time decision-making tasks compared to traditional RL. For example, [28] demonstrates the application of DRL in collision-free path planning against surrounding obstacles.

While DRL excels in adaptive decision-making for complex and dynamic environments, its generalization remains limited when facing significantly different scenarios from its training context. DRL demonstrates flexibility under local variations, but notable environmental shifts typically necessitate updated training. This challenge highlights the potential of advanced approaches, such as transfer learning, to improve generalization across diverse conditions, which is a promising avenue for future exploration beyond the scope of this thesis. Conversely, model-based methods often achieve superior performance in constrained environments but rely heavily on predefined parameters. These parameters require fine-tuning when the scenario changes, either through human intervention or environmental variations. Therefore, while DRL demands retraining for novel environments, model-based approaches also face adaptation challenges due to their dependency on scenario-specific configurations.

The DRL-based autonomous driving system is illustrated in Fig. 1.7. The agent interacts with the environment through actuators, observations, and rewards. The agent comprises a decision network that receives information from observations of the environment and uses rewards to assess its actions. These observations are provided by the observer, which interprets the state of the AV and its environment. Based on the observations, the agent generates control commands and then sends the commands to actuators. Following the actuation of these control commands, the renewed environment information and AV state are updated. Simultaneously, a reward function evaluates the agent's actions based on predefined metrics such as safety, efficiency, or compliance to driving norms. This reward function assigns positive or negative rewards depending on how well the AV's actions align with the desired outcomes. These rewards are then fed back to the agent, guiding the learning towards the optimal driving behavior.

DRL has been proven effective in handling emergency situations, which are critical for real-world driving scenarios. For example, [29] proposes a DRL-powered driving system designed to avoid collisions in emergencies. This system learns to react swiftly and safely to sudden changes, improving the robustness of decision-making in real-world conditions. Recently, several studies have been demonstrated in various scenarios [30–39]. However, different scenarios present distinct driving requirements, necessitating tailored algorithms. On highways, the decision-making of AVs primarily focuses on avoiding collisions with HDVs while maintaining a high average speed. In contrast, ramps introduce additional challenges, such as blocked areas that are not present on highways. Furthermore, it is essential to assess DRL-based algorithms based on demands from various social perspectives, including vehicle users, vehicle manufacturers, and public traffic systems. Research on DRL-based algorithms, categorized by driving scenarios and evaluated based on their adaptability to real-world demands, is crucial for identifying valuable research directions.

While DRL has long driven innovations in autonomous driving, recent studies have demonstrated that large-scale foundation models can significantly enhance various aspects of these systems. For instance, [40] provides a comprehensive survey on how these models can improve perception and decision-making, while also addressing key challenges such as computational efficiency and the sim-to-real gap. [41] introduces a language-based agent that integrates a large language model into conventional driving modules for high-level planning and reasoning. Similarly, [42] leverages language models to formulate driving as a model-predictive control problem, thereby enhancing safety and interpretability in complex scenarios. In addition, [43] proposes EMMA, a multimodal model that unifies vision and language inputs to jointly tackle perception and planning tasks, and [44] presents DriveDreamer, which utilizes a diffusion-based generative world model trained on real driving data to bridge the gap between simulation and real-world environments. Collectively, these works illustrate the transformative potential of large-scale foundation models for achieving robust, scalable, and safe autonomous driving systems.

1.5 Dissertation Structure

The dissertation consists of five chapters, with the outline as follows:

Chapter 1 introduces the motivation of this thesis. In addition, it analyzes and summarizes the challenges in autonomous driving and the advantages of using RL-based methods to address these challenges. Furthermore, a brief introduction of the research content is provided. Chapter 2 delves into the details of designing an appropriate RL-based framework for typical driving scenarios. Chapter 2 also identifies the research gaps and outlines further developments needed in current research. Chapters 3 and 4 address the challenges in autonomous driving based on the RL-based design rules defined in Chapter 2. In Chapter 3, an RL-based framework for highway driving is proposed, including driving safety [45, 46], driving efficiency [47, 48], training efficiency [36, 49], unselfishness [50–52], and interpretability [53, 54]. Simulation results demonstrate that the proposed framework outperforms other popular DRL algorithms. The environment considered in Chapter 4 is extended to autonomous racing, which is an extreme edge case of autonomous driving, requiring precise and safe control at very high velocities. The RL-based framework is adapted using DDTUI principles, resulting in significant improvements compared to current studies in autonomous racing. Simulation results on a physical engine demonstrate that the proposed algorithm achieves fewer collisions, higher peak rewards, reduced training time, and shorter lap times across multiple testing racetracks compared to benchmark algorithms. Chapter 5 concludes the study by summarizing key achievements, analyzing limitations, and outlining future research directions for RL-based autonomous driving.

Chapter 2

Literature Review

This chapter reviews DRL-based algorithms for autonomous interactive driving, classified by scenarios and evaluated for adaptation to real-world conditions. Four typical scenarios are included: highways, on-ramping merging, roundabouts, and unsignalized intersections. DRL-based decision-making approaches are reviewed for the four typical scenarios and evaluated using the criteria of driving safety, driving efficiency, training efficiency, unselfishness, and interpretability. The evaluation is consistent across all papers by examining the inclusion of evaluation factors in the designed algorithms and their corresponding verifications. For example, if a paper discusses safety but doesn't include verifications like a lower number of collisions or consistently maintaining safe distances d_s, it would not be considered to include the safety factor.

2.1 Road Features and Driving Tasks

This section provides the road features and driving tasks for AVs in the scenarios of highways, on-ramping merging, roundabouts, and unsignalized intersections.

2.1.1 Highways

2.1.1.1 Road Features and Driving Tasks

Highways are fundamental components of road networks, designed to enable vehicle movement over long distances with minimal interruption. The design of highways focuses on safety, efficiency, and environmental impact. Safety features include wide lanes and clear signage to reduce collision risks.High efficiency is achieved by optimizing lane layouts

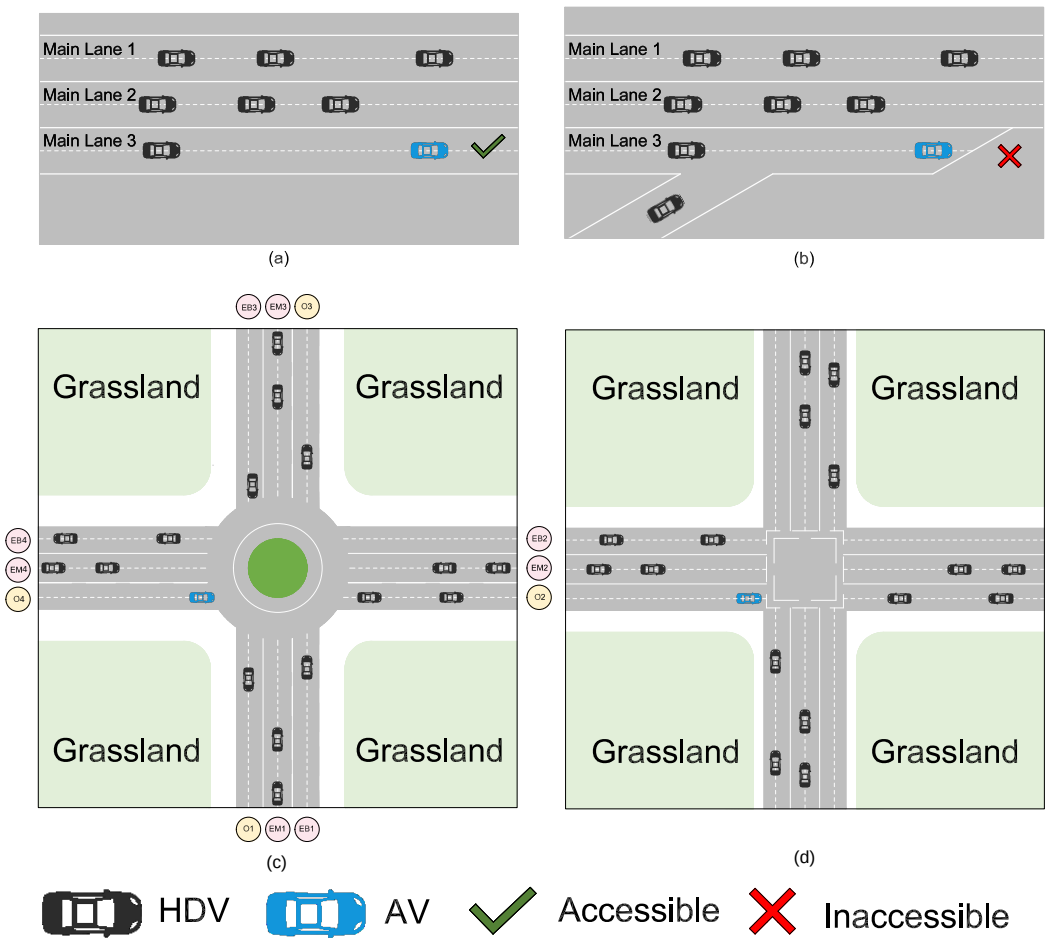


Figure 2.1: Example scenarios of autonomous driving: (a) highway; (b) on-ramp merging; (c) roundabout with 12 ports (8 entrances: EM1–EM4, EB1–EB4; 4 exits: O1–O4) and a central planted island; (d) unsignalized intersection.

to keep vehicles driving smoothly and reduce bottlenecks. The impact of highways on natural landscapes is reduced through careful route planning. The *M8 Motorway* in Glasgow is a major transport route connecting Glasgow and Edinburgh, known worldwide for its heavy traffic flow and complex junctions [55]. The *Interstate Highway System* in the United States is a vast network of highways designed to support long-distance travel and economic connectivity across states [56]. Similarly, Germany's *Autobahn*, known for its sections without speed limits, exemplifies the balance between high-speed travel and safety on highways [57].

2.1.1.2 An Example of a Highway

Fig. 2.1(a) presents a scenario involving a three-lane highway. The AV drives in main lane 3 and interacts with HDVs in all three lanes. There are no disturbances or uncertainties other than the surrounding HDVs. Therefore, the issue of driving safety primarily relates to collisions with surrounding HDVs. In the car-following phase, the AV can follow the HDV ahead by adjusting its acceleration. However, cautious following can lead to a loss of driving efficiency. To maintain high driving efficiency, the AV may change lanes when the space ahead is limited. However, collisions with HDVs in the target lane could occur during the lane-changing. Therefore, the driving task on highways can be summarized as balancing collision avoidance with surrounding HDVs while maintaining a consistently high speed.

2.1.2 On-ramping Merging

2.1.2.1 Road Features and Driving Tasks

Ramps, including on-ramps or off-ramps, are essential components of highway systems. Due to the symmetry between on-ramping and off-ramping processes, this chapter considers only on-ramping merging. Ramps enable the smooth and safe transition of vehicles between different roadways, typically connecting surface streets with highways. Ramps provide access to highways without disrupting traffic flow on the main highway lanes.

Ramp design focuses on safety, efficiency, and space utilization. Safety is crucial, as ramps must accommodate vehicles accelerating or decelerating while merging onto or diverging from the highway. On-ramps enhance traffic flow by reducing disruptions to mainline traffic and providing sufficient space for safe merging. Additionally, urban space constraints often require innovative ramp designs, such as cloverleaf interchanges, to connect multiple roadways effectively.

2.1.2.2 Comparison with Highways

Highways and ramps serve different functions, which are summarized below.

- Functionality: Highways are designed for high-speed, long-distance travel with minimal interruption, while ramps are the transition between different road types.
- Design: Highways are characterized by long, straight stretches with multiple lanes, designed to maintain high speeds and efficient traffic flow. In contrast, ramps often involve curves and elevation changes, designed to accommodate vehicles as they speed up or slow down.
- Speed: Highways support higher speeds, with vehicles typically traveling at constant high speeds over long distances. Ramps involve acceleration or deceleration, requiring careful design to manage the speed differential between the ramp lane and the main lane.

For example, the *Cloverleaf Interchange* is a common design that efficiently manages space while connecting highways with multiple surface streets [58]. Another example is the *High Occupancy Vehicle (HOV) lane ramps*, which are designed to control the flow of carpool vehicles onto highways, providing direct and less congested access points [59].

Consider a three-lane ramp scenario in Fig. 2.1(b), which includes two main lanes and one ramp lane. The AV interacts with both dynamic and static objects. The dynamic objects are surrounding HDVs, each with unique driving intentions, speeds, and acceleration patterns. The static object represents an obstruction within the ramp lane, rendering the lane impassable and blocking access. As a result, the AV must change into the main lane before the ramp ends, considering the HDVs and the feasibility in lane-changing.

Waiting for enough space to change lanes and driving slowly to avoid blocked roads lead to safer driving. However, this cautious driving can significantly reduce driving efficiency and lower road capacity on the ramp. Consequently, it is challenging to navigate the ramp, avoid collisions with both surrounding HDVs and the blocked road ahead, while still maintaining a high driving speed.

2.1.3 Roundabouts

2.1.3.1 Road Features and Driving Tasks

Roundabouts are designed to improve traffic flow and enhance safety by reducing the likelihood of severe accidents. One example of a typical roundabout is *Folon's obelisk in Pietrasanta* in Italy, which features a central island and circular roads around it [60]. Another example is the *Place Charles de Gaulle* in Paris, France, where twelve major avenues converge around the Arc de Triomphe [61].

2.1.3.2 An Example of a Roundabout

An example of a roundabout is presented in Fig. 2.1(c). The AV starts from the EB4 port and has three possible exit choices: O1, O2, and O3. When the target exit is O1, the AV can simply follow the outer lane. For the target exit O2, there are two possible routes. One route is staying in the outer lane, which is generally safer. The other route is merging into the inner lane and exiting near O2. This second route is more efficient, as the inner lane offers a shorter curve length for the same round angle. However, rear vehicles driving in the inner lane bring potential collision risks. For the target exit O3, the AV must find the right moment to merge into the inner lane. After traveling in the inner lane for a period, the AV is expected to change lanes near the exit. The main challenge is to safely interact with other HDVs when approaching each of these three exits.

2.1.4 Unsignalized Intersections

2.1.4.1 Road Features and Driving Tasks

Unsignalized intersections are critical components of road networks where two or roads meet or cross. They are designed to manage traffic flow from different directions, enabling vehicles to navigate safely through crossing points. Unsignalized intersection can control and organize traffic movements, reduce congestion, and enhance safety for all vehicles. One example is the *Diverging Diamond Interchange (DDI)* [62].

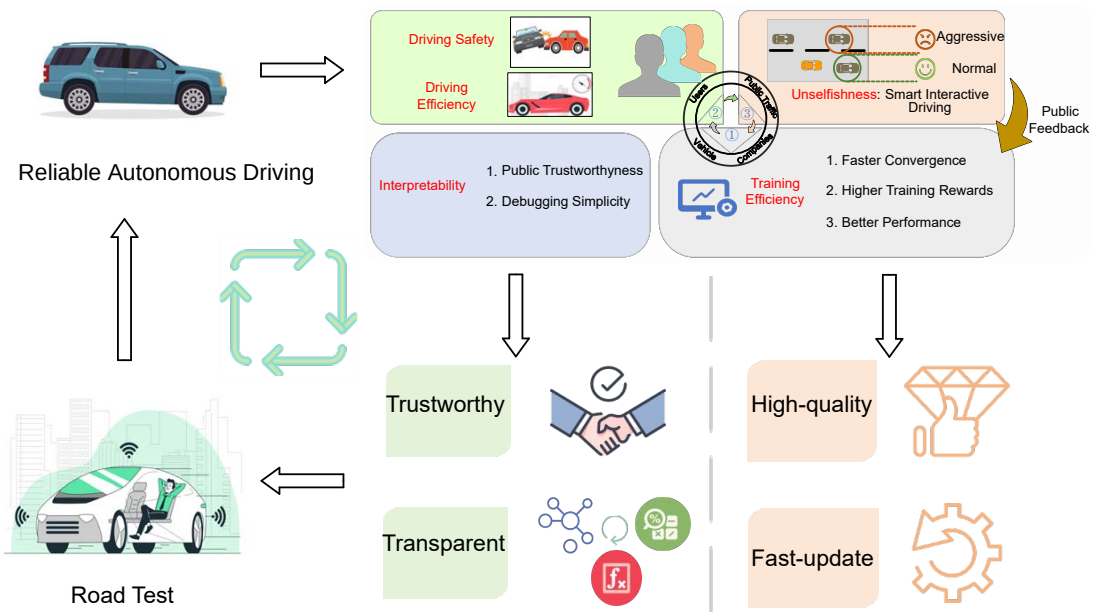


Figure 2.2: The importance and necessaries of achieving DDTUI in real-world autonomous driving.

2.1.4.2 An Example of an Unsignalized Intersection

Fig. 2.1(d) shows a three-lane unsignalized intersection designed for moderate to heavy traffic flow. The intersection accommodates vehicles from all four directions, with dedicated lanes for specific traffic movements. Each approach to the intersection includes three lanes, and the areas surrounding the intersection are grassland. At the center, where all four roads meet, there is an ample space for vehicles to make turns from any direction. This central area is essential for preventing bottlenecks and ensuring smooth traffic flow.

2.2 Rationale of the Evaluation Factors

In the context of adapting decision-making algorithms to real-world driving, five key evaluation factors have been selected: driving safety and efficiency, training efficiency, unselfishness, and interpretability. As depicted in Fig. 2.2, driving safety and efficiency form the foundation of any autonomous driving system. Training efficiency enables faster convergence of algorithms. Unselfishness enhances interaction with surrounding traffic, promoting cooperation with HDVs. Meanwhile, interpretability fosters public trust and addresses algorithmic errors, ensuring that decision-making is transparent and understandable. The detailed rationale behind selecting these factors is discussed below.

2.2.1 Driving Safety

Driving safety is a fundamental requirement for autonomous vehicles. Frequent collisions cause substantial economic losses and pose severe safety risks [63, 64]. Therefore, driving safety is primarily evaluated based on the frequency of collisions with other vehicles [65, 66]. Minimizing collisions is a direct measure of the vehicle's compliance to safety standards. Collision avoidance commonly relies on flexible reactions to hazardous areas. Once a hazard is detected, the system assesses the risk by analyzing the relative speed, distance, and trajectory of surrounding objects [67]. Additionally, some autonomous driving systems evaluate possible decisions to avoid collisions while maintaining high efficiency [68, 69]. Furthermore, other autonomous driving systems use rule-based commands to adjust the AV's behavior when unsafe conditions emerge [70]. For instance, AVs will be asked to stop when they encounter an interaction and spot-lines simultaneously [70].

2.2.2 Driving Efficiency

Driving efficiency refers to an AV's ability to maintain a high average speed while adapting to varying traffic conditions. However, the implications of driving efficiency extend far beyond speed, affecting road capacity, user experience, and energy consumption.

On road capacity, efficient driving allows vehicles to travel at optimal speeds, minimizing delays and reducing traffic congestion. For example, HDVs tend to drive faster on familiar roads, contributing to higher road capacity and traffic flow [71–73]. Similarly, AVs promote smoother traffic flow when they operate efficiently. Therefore, efficient driving allows more vehicles to travel smoothly without congestion. On user experience, an efficient journey means shorter travel time and a smoother ride, significantly improving overall satisfaction [74–76]. Besides, improving driving efficiency is crucial for reducing the energy consumption [77, 78].

2.2.3 Training Efficiency

Training efficiency of algorithms directly impact the time and resources required to bring a fully functional AV system to reality. One primary benefit of improved training efficiency is the reduced training time. The acceleration allows developers to focus more on system fine-tuning and extensive testing. Several studies have reduced training time by adding extra training mechanisms or adjusting the structures of networks [36, 79–81]. Another important benefit is the reduction in device wear and tear. Fast and efficient training reduces the required computational resources. By improving training efficiency, the workload of computing equipment is minimized, resulting in less frequent maintenance and replacement.

2.2.4 Unselfishness

In the context of autonomous driving, unselfishness refers to an AV's ability to consider and accommodate the intentions of other HDVs on the road. Unselfishness evaluates how well an AV can cooperate with surrounding vehicles by predicting their intentions and adjusting its behavior accordingly. Human drivers often prioritize factors such as safety, efficiency, and comfort, and these intentions vary widely depending on the specific situations.

Accurately classifying these driving intentions is essential for effective interactions with surrounding HDVs. Existing methods for recognizing driving intentions and enabling interaction-aware driving have been reviewed in [82]. These methods categorize driving intentions across various scenarios, including car following and lane changing [83,84]. While many papers have focused on the self-driving characteristics of the ego vehicle [85,86], the importance of unselfish behavior is becoming increasingly recognized.

An unselfish AV that effectively anticipates and responds to the intentions of other vehicles contributes to a smoother and more harmonious traffic flow. By avoiding overly aggressive or excessively cautious driving behaviors, the AV can help minimize disruptions and conflicts with other vehicles. This cooperative approach enhances the safety and efficiency of all vehicles in the road network and improves the driving experience for everyone involved.

2.2.5 Algorithm Interpretability

Algorithm interpretability has gained significant importance due to DRL models are required to make logical decisions. A logical structure makes the black-box of learning more transparent [53, 54]. In DRL-based autonomous driving systems, improving interpretability is crucial for system's safety and transparency. To address the challenges in interpretability, various approaches have been adopted, including policy visualization to showcase DRL behaviors [87,88], and surrogate models for approximate human-understandable explanations [89–92]. Furthermore, specific rule-based methods, algorithmic structure-adapted methods, and human-grounded methods have been proposed to assess interpretability.

Specific rules have been developed to assess interpretability [93]. One such rule, known as *FAST*, evaluates interpretability via four criteria: *F* for fairness, *A* for accountability, *S* for sustainability, and *T* for transparency [94]. Fairness requires models to be formalized using basic explanation labels and functionality evaluation. Accountability refers to answerability and auditability, ensuring that the system has been clearly defined. Sustainability ensures safe operation without inequality or discrimination, while transparency ensures that the model's internal rule settings are accessible and understandable.

Some methods focuses on assessing interpretability by adjusting DRL algorithmic structures. Researchers achieve this by developing standardized benchmarks that use interpretability metrics [95,96] or by troubleshooting explanations [97,98] to identify instances where these explanations fall short. In addition, some studies concentrate on altering neural network architectures to enhance interpretability [99, 100].

Furthermore, human-grounded methods focus on how easily people can understand the model's key computational sections [101]. DRL-based algorithms incorporating traffic-related models enable people to better understand their structures through traffic knowledge or mathematical formulation, thereby improving interpretability.

2.3 Deep Reinforcement Learning-based decision-making on Highways

2.3.1 Single-factor Methods for Highway Driving

Many works consider only one of five key factors. A Double Deep Q-Network (DDQN) is integrated with handcrafted safety and dynamically-learned safety modules in [102]. The handcrafted safety module relies on heuristic safety rules derived from common driving practices, ensuring a safety distance, d_s, with other vehicles. The dynamically-learned safety module uses driving data to learn safety patterns. By integrating both the handcrafted and dynamically-learned safety modules, the driving safety is improved.

Moreover, deep deterministic policy gradients (DDPG) have been used to improve driving efficiency by overtaking surrounding vehicles in [103]. The overtaking-oriented training is achieved by adding a high reward for overtaking maneuvers. The reward function for overtaking is formulated as [103]:

$$R_{\text{overtaking}} = R_{\text{lane\_keeping}} + 100 \times (n - P_r) \tag{2.1}$$

where $R_{\text{lane\_keeping}}$ is the reward for lane-keeping, n is the total number of vehicles in a given episode, and P_r reflects the number of vehicles in front of the AV. Therefore, the larger the P_r, the smaller the $R_{\text{overtaking}}$. Although safety rewards are applied, the collision rate increases with the frequency of overtaking.

Furthermore, non-linear model predictive control (NMPC) has been integrated with DDQN to maintain safe highway driving in [104]. NMPC inherently incorporates vehicle dynamics as constraints into its optimization, ensuring that the control inputs from the DRL agent remain within safe and feasible bounds [104]:

$$\begin{aligned}
&\min_{x(t),u(t)} \int_0^T e(t)^\top Q e(t) + r\delta^2(t) + ru^2(t)\mathrm{d}t \\
&\text{s.t.} \quad \dot{x}(t) = f(x(t), u(t)), \\
&\qquad e_{y_{\min}} \leq e_y(t) \leq e_{y_{\max}}, \\
&\qquad e_{\psi_{\min}} \leq e_\psi(t) \leq e_{\psi_{\max}}, \\
&\qquad \delta_{\min} \leq \delta(t) \leq \delta_{\max}, \\
&\qquad u_{\min} \leq u_1(t) \leq u_{\max}
\end{aligned} \tag{2.2}$$

where T is the prediction horizon, $e(t)$ is the error vector to be regulated to zero, and $Q = \mathrm{diag}(q_1, q_2)$ is a diagonal matrix of tracking weights. The control effort weight is denoted by r. The steering angle is represented by $\delta(t)$, and the control input is $u(t)$. The state vector is $x(t)$, and f represents the system dynamics. $e_y(t)$ and $e_\psi(t)$ are the lateral position error and heading angle error, respectively. The variables $e_{y_{\min}}$, $e_{y_{\max}}$, $e_{\psi_{\min}}$, $e_{\psi_{\max}}$, $\delta_{\min}$, $\delta_{\max}$, $u_{\min}$, and $u_{\max}$ are the minimum and maximum admissible values for the lateral position error, heading angle error, steering angle, and control input, respectively. NMPC improves the interpretability of safe control by providing a clear mathematical formulation that integrates the system's constraints with the agent's decision-making [105–107].

Additionally, a policy gradient (PG) method has been used with hard constraints to ensure safe highway driving in [108]. These hard constraints prevent the AV from approaching risky boundaries, such as track edges. For example, the AV's longitudinal and lateral positions are restricted from approaching the track boundaries. Cooperative lane-changing has been achieved in [109], enhancing the unselfishness. Interpretability has been improved by combining DRL with imitation learning (IL) in [110]. IL uses expert demonstrations to make the learning more interpretable. Training efficiency in highway driving is also enhanced by integrating a spatial attention module and attention mechanism into the deep Q-network in [111].

2.3.2 Dual-factor Methods for Highway Driving

Additionally, two of the five considered factors are integrated in some recent studies. The Intelligent Driver Model (IDM) [112] has been incorporated into the DDQN for highway driving in [113]. The IDM prevents collisions during car-following and therefore, the integration of DDQN with IDM enhances both the driving safety and interpretability in highway driving. The IDM is formulated as [113]:

$$U_{\mathrm{IDM}} = U_{\max}\left[1-\left(\frac{v_{\mathrm{FV}}}{v_e}\right)^4-\left(\frac{g^*}{g}\right)^2\right] \tag{2.3}$$

where $U_{\max}$ is the maximum acceleration of the AV, v_e is the expected velocity, and g is the gap between the AV and the HDV. The desired gap g^* between the AV and the front HDV is formulated as [113]:

$$g^* = d_s + v_{\mathrm{AV}}T_e - \frac{v_{\mathrm{AV}}\Delta v}{2\sqrt{U_{\max}b}} \tag{2.4}$$

where T_e is the expected time gap, Δv is the velocity difference between the AV and the front vehicle (FV), and b is the comfortable deceleration.

The reward function of DDQN has been adapted to improve driving safety and efficiency in [114]. Specifically, a penalty is applied when the vehicle goes off-road or the time-to-collision (TTC) falls below a threshold [115] . The reward for driving efficiency is formulated as [114]:

$$R = \frac{v_0}{v_{\max}} \tag{2.5}$$

where $v_{\max}$ is the maximum velocity, and v_0 is the current velocity. This reward function helps maintain a relatively high driving velocity, thus increasing driving efficiency. Moreover, driving safety and altruism have been achieved using a level-k game-based DQN in [116]. The level-k game models the reasoning interaction between AVs and HDVs, promoting unselfish decision-making. A crash penalty is implemented in the DQN to prevent frequent collisions between AVs and HDVs. Additionally, unselfishness and training efficiency have been considered in [117]. Unselfishness is achieved through a cooperative

multi-goal credit function-based policy gradient (PG). This adapted PG accounts for the goals of all vehicles, optimizing overall performance during training. Training efficiency is improved by a multi-agent reinforcement learning (MARL) curriculum, which reduces the number of trainable parameters and lowers computational costs.

MARL plays a critical role in promoting unselfishness by enabling agents to learn cooperative behaviors through shared decision-making. It allows autonomous vehicles to consider the intentions of surrounding agents, fostering smoother traffic flow and safer interactions. This cooperative learning aligns closely with game theory principles, where each agent behaves like a player in a multi-agent system, adjusting strategies based on others' actions. While traditional game theory relies on predefined equilibrium models, MARL enhances this by dynamically learning optimal policies through continuous interaction. In this thesis, game theory has been integrated as part of the unselfish driving framework. By applying game theory models, the decision-making model ensures that autonomous vehicles prioritize both their efficiency and the well-being of surrounding road users, promoting cooperative and socially aware driving behavior.

Unselfishness and driving efficiency on highways are achieved in [118]. Unselfishness is promoted through MARL by considering each vehicle's state. Driving efficiency is enhanced by a reward function that selects actions to increase the average velocity of all vehicles. Driving safety and driving efficiency have been achieved using multi-objective approximate policy iteration (MO-API) in [119]. Driving safety is ensured by monitoring collisions, while driving efficiency has been assessed by comparing the v_0 with the v_e. In [120], driving efficiency and unselfishness are considered in highway driving. Driving efficiency is achieved by a reward based on v_0, $v_{\max}$, and $v_{\min}$. Unselfishness is achieved by penalizing unnecessary lane changes to reduce disturbances to HDVs. Driving safety and training efficiency have been addressed in [121]. Safety is maintained by ensuring a d_s between vehicles using rule-based constraints, while training efficiency is improved by incorporating a multi-head attention mechanism.

2.3.3 Three-factor Methods for Highway Driving

Furthermore, three of the five considered factors are combined in a few recent papers. Driving safety, interpretability, and driving efficiency have been improved in [122]. Driving safety and interpretability are enhanced by using a collision penalty and the IDM. Driving efficiency is ensured by a reward based on the velocity difference between $v_{\max}$ and v_0. The reward at time step t is formulated as [122]:

$$R_t = -\text{Collision} - 0.1 \times (v^t_{\max} - v^t_0) - 0.4 \times (L-1)^2 \tag{2.6}$$

where Collision, $v^t_{\max}$, and v^t_0 are the occurrence of a collision, maximum velocity, and current velocity at time t, respectively. L represents the relative position the target lane, where $L = 1$ indicates that the vehicle has successfully reached the target lane. A collision results in a negative reward, and a larger difference between v_0 and $v_{\max}$ also leads to a negative reward. Additionally, if the vehicle does not drive in the target lane, a penalty is applied.

A multi reward-based DQN has been proposed to achieve safe, efficient, and unselfish driving in [123]. Three rewards are combined: speed reward, limited lane-changing reward, and overtaking reward. The speed reward is a normalized reward based on the current speed relative to the minimum and maximum speed limits [123]:

$$R_v = \frac{(v_0 - v_{\min}) * r_v}{v_{\max} - v_{\min}} \tag{2.7}$$

where R_v represents the reward for speed, encouraging higher speeds within safe limits. $v_{\min}$ is the minimum speed of the agent vehicle, and r_v is the base reward for speed. The limited lane-changing reward function is designed to minimize the number of lane changes, promoting safer driving and reducing the disturbance to surrounding vehicles:

$$R_l = \begin{cases} -r_l, & \text{if the agent vehicle changes lanes;} \\ 0, & \text{otherwise.} \end{cases} \tag{2.8}$$

where $-r_l$ is the penalty value for a lane change. The overtaking reward function encourages the agent vehicle to overtake more vehicles, improving driving efficiency:

$$R_o = \begin{cases} r_o, & \text{if the agent vehicle overtakes another vehicle;} \\ 0, & \text{otherwise.} \end{cases} \tag{2.9}$$

where r_o is the reward value for overtaking.

Interpretability, driving safety, and driving efficiency have been achieved in [124]. Safety and efficiency are enhanced by penalizing frequent lane changes and tracking the desired velocity v_d, respectively. Interpretability is achieved through a car-following process using a proportional-derivative (PD) controller with transparent mathematical formulation [124]:

$$d_{\text{des},i} = \alpha_i^j v_l^{j+1} \tag{2.10}$$

$$a_{\text{cf},i} = K_p(x_l^{j+1} - x_i^j) + K_d(v_l^{j+1} - v_i^j) \tag{2.11}$$

where $d_{\text{des},i}$ is the desired following distance for the i-th vehicle, α_i^j is a sensitivity parameter with random values from $\mathcal{N}(1.3, 0.02)$, v_l^{j+1} is the speed of the leading vehicle in the $(j+1)$-th lane, $a_{\text{cf},i}$ is the acceleration command, K_p and K_d are the proportional and derivative gains, and x_l^{j+1} and x_i^j are the positions of the leading and i-th vehicles, respectively.

In [125], driving safety, efficiency, and interpretability have also been combined. Safety and efficiency are achieved by penalizing collisions and rewarding high average velocity. Interpretability is enhanced by using the risk potential field (RPF), which models and visualizes risks around surrounding vehicles. In [126], driving safety and interpretability have been achieved in adaptive cruise control (ACC), which maintains d_s between vehicles and provides interpretable mathematical formulations. Driving efficiency is achieved by rewarding each high-speed state. Finally, driving safety, driving efficiency, and training efficiency have been achieved in [127]. Safety is ensured through a collision penalty, and efficiency is rewarded based on the velocity difference between v_0 and $v_{\min}$. Training efficiency is improved by using a long short-term memory (LSTM) network-assisted DDQN. However, compared with approaches such as [122], which rewards efficiency based on the

velocity gap between $v_{\max}$ and v_0 while also integrating safety and interpretability via collision penalties and lane deviation measures, and [124], where a PD controller provides clear mathematical transparency for interpretability, the method in [127] uniquely emphasizes rapid training convergence through its DDQN enhanced with LSTM. Moreover, while [125] utilizes a RPF to offer an intuitive visualization of surrounding hazards, and [126] leverages ACC to ensure a fixed safe distance with explicit formulations, the approach in [127] may require further exploration regarding its robustness and stability in diverse and dynamic driving scenarios, given the additional complexity introduced by the LSTM component.

2.3.4 Four-factor Methods for Highway Driving

Moreover, four of the five considered factors have been included in some studies. Driving safety, driving efficiency, training efficiency, and interpretability have been considered in [128]. Driving safety and driving efficiency are achieved by a reward function that maintains a d_s from the leading vehicle while tracking the v_d. Interpretability is ensured through safety-based driving rules [128]:

$$t_{f_{\min}} = \inf\left\{t : t > \frac{2(v - v_{f_{\text{target}}})}{a_{d_{\max}}}\right\} \tag{2.12}$$

$$t_{b_{\min}} = \inf\left\{t : t > \frac{2(v_{b_{\text{target}}} - v)}{a_{d_{\max}}}\right\} \tag{2.13}$$

$$d_{\text{target}_{\min}} = \min\left\{\frac{(v - v_f)t_f}{2}, \frac{(v_b - v)t_b}{2}\right\} \tag{2.14}$$

$$\Delta d_{\text{target}} = \min\left\{|x_{\text{AV}} - x_{f_{\text{target}}}|, |x_{\text{AV}} - x_{b_{\text{target}}}|\right\} \tag{2.15}$$

where $t_{f_{\min}}$ and $t_{b_{\min}}$ are the minimum safe time intervals between the AV and the vehicles in front and behind in the target lane, respectively. v is the speed of the AV; $v_{f_{\text{target}}}$ and $v_{b_{\text{target}}}$ are the speeds of the front and behind vehicles in the target lane, respectively. $a_{d_{\max}}$ is the maximum deceleration. $d_{\text{target}_{\min}}$ is the minimum distance between the AV and the FV in the target lane, and Δd_{target} is the actual distance between the AV and the nearest vehicle in the target lane. x_{AV}, $x_{f_{\text{target}}}$, and $x_{b_{\text{target}}}$ represent the horizontal coordinates of the AV, the front target vehicle, and the vehicle behind in the target lane,

respectively. By implementing these safety rules, the decision-making of the AV becomes more transparent and interpretable. Training efficiency is achieved through the potential-based reward shaping function. The total reward function and reward shaping function are given by [128]:

$$R' = R(s,a,s') + \beta F(s,a,s') \tag{2.16}$$

$$F(s,a,s') = \gamma\phi(s') - \phi(s) \tag{2.17}$$

$$F(s,a,s',t,t') = \gamma\phi(s',t') - \phi(s,t) \tag{2.18}$$

where R' is the new reward criterion, $R(s,a,s')$ is the original reward function, β is a weighting factor, $F(s,a,s')$ is the potential-based reward shaping function, s and s' are the current and next state, respectively, a is the action taken, and γ is the discount factor. $\phi(s)$ is the potential function mapping the state to a real number, and t and t' are the time corresponding to s and s', respectively. (2.16) combines the original reward function with an additional shaping term. (2.17) defines the shaping function as the difference between the discounted potential of the next state and the current state. (2.18) extends (2.17) by including time as a parameter and therefore allows for dynamic potential functions.

Driving safety, driving efficiency, unselfishness, and training efficiency on highways have been addressed in [129]. Driving safety and efficiency are considered in the reward function of the DQN. Unselfishness is achieved through a joint policy update, accounting for the profits of multiple vehicles. Training efficiency is enhanced by reusing the experiences of single agents within a MARL framework. In [130], driving safety, efficiency, unselfishness, and training efficiency on highways have been explored. Safety and efficiency are ensured by assessing the remaining reaction time during emergencies and selecting the proper lane-changing point, respectively. Unselfishness is achieved using MARL for cooperative highway driving, while training efficiency is improved with a dynamic coordinate graph (DCG) that enhances cooperative efficiency. In [131], safety, efficiency, unselfishness, and training efficiency have been considered. Safety is ensured by applying penalties both for

collisions and for deviating from the road. Efficiency is achieved by rewarding each state that overtakes other vehicles. Unselfishness is promoted through MARL to coordinate driving. Training efficiency is enhanced by employing a parameter-sharing mechanism, which stores experience of each agent to reinforce common scenario understanding.

In [132], safety, efficiency, unselfishness, and interpretability have been considered. Safety, efficiency, and unselfishness are improved through rewards for collisions, velocity ratio between v_0, $v_{\max}$, and $v_{\min}$, and limiting unnecessary lane changes, respectively. Interpretability is enhanced by integrating an autonomous emergency braking system, promoting safer decision-making. In [133], safety, efficiency, interpretability, and training efficiency have been addressed. Safety and efficiency are enhanced by adding a safety layer and incorporating the ratio between longitudinal speed, $v_{\max}$, and $v_{\min}$. Interpretability is improved by using a support vector machine (SVM), which provides interpretable safe decision boundaries. Training efficiency is boosted through an external space attention mechanism that pays attention to the crucial areas of surrounding environment.

In [134], safety, efficiency, unselfishness, and training efficiency have been tackled. Safety, efficiency, and unselfishness are achieved through rewards for collisions, velocity ratios, and MARL, while training efficiency is improved using a distributional DQN with multi-type input data. Finally, in [135], safety, efficiency, unselfishness, and interpretability have been considered. Safety, efficiency, and unselfishness are enhanced through rewards for collisions, target velocity differences, and unnecessary lane changes, respectively. Interpretability is achieved through rule-based constraints, such as preventing lane changes with short lateral distances to lead vehicles.

2.3.5 Five-factor Methods for Highway Driving

Additionally, all the five factors are addressed in some studies, such as [136]. Driving safety and efficiency, and unselfishness are achieved by reducing collisions, increasing speed, and minimizing lane-change frequency through rewards. Training efficiency is improved through a convolutional neural network-based LSTM. Interpretability is enhanced by

Table 2.1
Evaluation of the DRL-based decision making in highway driving

Reference	Year	Safety	Efficiency	Training Efficiency	Unselfishness	Interpretability
[102]	2020	Safety modules	-	-	-	-
[103]	2018	-	Overtaking reward	-	-	-
[104]	2023	NMPC constraints	-	-	-	-
[108]	2016	Hard constraints	-	-	-	-
[109]	2019	-	-	-	Local interactions	-
[110]	2023	-	-	-	-	Imitation learning
[111]	2020	-	-	Attention module	-	-
[113]	2019	IDM integration	-	-	-	IDM integration
[114]	2020	TTC threshold	Velocity reward	-	-	-
[116]	2021	Crash penalty	-	-	Level-k game	-
[117]	2018	-	-	MARL curriculum	Cooperative function	-
[118]	2018	-	Average velocity	-	MARL	-
[119]	2018	Collision monitoring	Velocity comparison	-	-	-
[120]	2019	-	Velocity reward	-	Lane change penalty	-
[121]	2022	Rule-based	-	Attention mechanism	-	-
[122]	2020	IDM & collision	Velocity difference	-	-	IDM integration
[123]	2019	Speed-limit reward	Overtaking reward	-	Lane-change limit	-
[124]	2018	Lane change penalty	Velocity tracking	-	-	PD controller
[125]	2021	Reward function	Reward function	-	-	Risk potential field
[126]	2021	Adaptive cruise	High-speed reward	-	-	ACC formulations
[127]	2020	Collision penalty	Velocity difference	LSTM-DDQN	-	-
[128]	2022	Safety rules	Reward function	Reward shaping	-	Safety rules
[129]	2019	Reward function	Reward function	MARL reuse	Joint policy	-
[130]	2018	Reaction time	Lane-changing point	DCG efficiency	MARL	-
[131]	2019	Collision penalties	Overtaking reward	Parameter sharing	MARL	-
[132]	2021	Collision rewards	Velocity ratio	-	Lane change limit	Emergency braking
[133]	2024	Safety layer	Velocity ratio	Attention mechanism	-	SVM boundaries
[134]	2019	Collision rewards	Velocity ratio	Distributional DQN	MARL	-
[135]	2022	Collision rewards	Velocity difference	-	Lane change penalty	Rule-based
[136]	2022	Collision reduction	Speed increase	CNN-LSTM	Lane change limit	Representations

'-' indicates that the corresponding factor was not explicitly addressed in the study.

using spatio-temporal image representations for HDVs, which increase the interpretability of the inputs. The DRL-based decision making in highway driving based on DDTUI is summarized in Table 2.1. In this thesis, "unselfishness" in autonomous driving includes cooperative driving as a model, exemplified by MARL and game theory methods.

2.4 Deep Reinforcement Learning-based decision-making in On-ramping Merging

2.4.1 Single-factor Methods for On-ramping Merging

Driving efficiency has been considered using Q-learning in [137]. The remaining time of AV on the ramp lane is reduced by optimizing the reward function, thus promoting fast lane-changing to the main lane. The reward function is formulated as [137]:

$$r_t = \mu \bar{v}_t + \omega \bar{q}_t, \quad \mu > 0, \quad \omega < 0 \tag{2.19}$$

where r_t represents the reward after taking action a_t; $\bar{v}_t$ denotes the average speed in the merging area during time step t; $\bar{q}_t$ indicates the average queue length at the on-ramp during time steps t and $t+1$; μ is a positive weight assigned to the speed reward, and ω is a negative weight for the queue length reward. These rewards help balance the trade-off between enhancing vehicle mobility on the mainline and reducing delays at the on-ramp. Driving efficiency has also been improved in [138] by reducing the total travel time reward (R_{TTT}), represented by the summation of the total number of vehicles at each time step. Driving safety has been achieved through a safety factor in [139]. The safety factor is a negative reward when the relative distances between AV and HDV are small. Driving safety has been achieved in [140], by giving rewards for each state having a d_s and penalties for collisions.

2.4.2 Dual-factor Methods for On-ramping Merging

Driving efficiency and unselfishness have been considered in [141]. Driving efficiency is achieved by using the average velocity of AVs as part of the reward, and unselfishness is achieved using MARL to maximize general profits. Interpretability and driving efficiency have been addressed in [142]. Interpretability is achieved by using DDPG to tune a tra-

ditional controller's parameters, keeping the traditional controller as the main system to ensure transparency. Driving efficiency is enhanced by reducing the error state, which reflects the gap between actual and critical traffic density. A smaller error state leads to higher traffic flow and average speed.

2.4.3 Three-factor Methods for On-ramping Merging

Driving efficiency, interpretability, and training efficiency have been addressed in [143]. Driving efficiency is achieved through a reward using the difference between the start and end time of each trip. Training efficiency is improved by a teacher-student model to train the decision-making system, where the traditional control method acts as the teacher guiding the DQN student. Similarly, driving efficiency, interpretability, and unselfishness have been improved in [144]. Driving efficiency is achieved by a reward that compares the average speed between two consecutive time. Unselfishness and interpretability are achieved by combining ramp metering (RM) with Q-learning. RM optimizes average vehicle speed and is algorithmically transparent. Driving safety, efficiency, and unselfishness have been addressed in [145]. Driving safety is achieved through a penalty for small relative distances, driving efficiency is enhanced by minimizing the relative distance while maintaining at least the safe distance, and unselfishness is achieved using MARL to optimize general driving performance.

2.4.4 Four-factor Methods for On-ramping Merging

Driving efficiency, training efficiency, unselfishness, and interpretability have been improved in [146], where driving and training efficiency is enhanced by DDPG-assisted RM and variable speed limit (VSL) control. Interpretability and unselfishness are improved through RM and VSL, which are algorithmically transparent. Driving safety, efficiency, training efficiency, and interpretability have been achieved in [147], where safety and interpretability are enhanced by combining APF, which quantifies and visualizes risk areas and provides interpretable input. Driving and training efficiency are achieved by combining MPC with DDQN, which outperforms single MPC or DDQN methods. Compared with this approach, [143] focuses on efficiency through a reward based on the time difference be-

tween the start and end of each trip and improves training efficiency via a teacher-student model in which traditional control guides DQN; however, it does not explicitly quantify risk areas or provide the visual interpretability offered by the APF in [147]. Similarly, [144] achieves efficiency by comparing average speeds over consecutive intervals and enhances interpretability through ramp metering combined with Q-learning, yet its safety mechanism is less comprehensive compared to the risk visualization provided by APF. In [145], driving safety is enforced by penalizing small relative distances and efficiency is maintained by ensuring a safe gap, but this method lacks advanced control techniques such as MPC and does not offer the same level of interpretability as APF-based approaches. Meanwhile, [146] integrates DDPG-assisted ramp metering and variable speed limit control to boost driving and training efficiency, along with unselfishness and interpretability; however, its mechanism does not explicitly provide the quantification and visualization of risks that characterize the approach in [147]. Collectively, while each method exhibits strengths in specific aspects, the approach in [147] stands out for its balanced integration of safety, efficiency, training efficiency, and interpretability, serving as a promising baseline for future research in on-ramping merging. Similarly, driving safety, efficiency, training efficiency, and unselfishness have been addressed in [148], where safety and efficiency are promoted by penalties for collisions and stop maneuvers. Training efficiency is improved by integrating the driver's intention model (DIM) with DDPG, while unselfishness is achieved by considering HDVs' various cooperation intentions. In [149], driving safety, efficiency, training efficiency, and interpretability have been achieved by applying the safety, efficiency rewards, and IDM respectively, with independant PPO (IPPO) used for improved training efficiency compared to baseline algorithms.

2.4.5 Five-factor Methods for On-ramping Merging

In [150], driving safety and efficiency have been achieved through collision and stable speed assessment rewards, respectively. Training efficiency is improved by using a safety supervisor, filtering detectable collision cases. Interpretability is enhanced through rule-based safety constraints, and unselfishness is achieved using MARL to maximize general profits. Similarly, all factors have been addressed in [151], where driving safety and efficiency are achieved via collision rewards and a velocity ratio, respectively. Training efficiency is

Table 2.2
Evaluation of the DRL-based decision making in on-ramping merging

Ref.	Year	Safety	Efficiency	Training Efficiency	Unselfishness	Interpretability
[137]	2021	-	Reward function	-	-	-
[138]	2019	-	Travel time reward	-	-	-
[139]	2017	Safety factor	-	-	-	-
[140]	2020	Collision-free driving	-	-	-	-
[141]	2019	-	Average velocity reward	-	MARL	-
[142]	2022	-	Error state reduction	-	-	Traditional controller
[145]	2022	Distance penalty	Distance minimization		MARL	-
[143]	2024	-	Trip time difference	Teacher-student model	-	Traditional control
[144]	2023	-	Speed comparison	-	Ramp metering	Ramp metering
[146]	2022	-	DDPG-assisted RM	DDPG	RM and VSL	RM and VSL
[147]	2024	APF	MPC with DDQN	MPC with DDQN	-	APF
[148]	2022	Collision penalty	Stop maneuver penalty	DIM with DDPG	HDV intentions	-
[149]	2023	Safety reward	Efficiency reward	IPPO	-	Rule-based
[150]	2023	Crash evaluation	Stable speed assessment	Safety supervisor	MARL	Rule-based constraints
[151]	2023	Collision rewards	Velocity ratio	Adversarial constraints	Nash-based game	Transparent game process
[152]	2023	DRAC	Velocity ratio reward	Multi-state rep.	Vehicle coop.	DRAC

enhanced by adversarial constraints, while unselfishness and interpretability is enhanced through a transparent Nash-based game that considers HDV's profits. Finally, in [152], driving safety and interpretability have been achieved using the deceleration rate to avoid a crash (DRAC), which has a detailed mathematical formulation and is transparent. Driving efficiency is improved by using (5) as an efficiency reward. Unselfishness is addressed by considering the cooperation intentions of other vehicles, and training efficiency is improved using multi-state representations to enhance the agent's learning capabilities. The DRL-based decision making on on-ramp merging based on DDTUI is summarized in Table 2.2.

2.5 Deep Reinforcement Learning-based decision-making at Roundabouts

2.5.1 Single-factor Methods for Roundabout Driving

Driving efficiency in roundabout driving has been improved using soft actor-critic (SAC) with higher peak rewards in [153]. Training efficiency has been achieved through action repeat and asynchronous advantage in [154]. Action repeat improves efficiency by allowing the agent to repeat the same action for several time steps, decreasing the frequency of making new decisions. Asynchronous advantage enables each agent to share its interaction experience with others. Training efficiency has been further improved by embedding the operational design domain (ODD) into DQN in [155]. ODD guides the training to more targeted scenarios, reducing unnecessary exploration and accelerating convergence.

2.5.2 Dual-factor Methods for Roundabout Driving

Driving efficiency and training efficiency are improved using the Conditional Representation Model (CRM) in [156], which helps the agent better understand safety by defining each state as safe or unsafe state. Training efficiency and interpretability have been improved by leveraging labeled data from domain experts as guidance in [157]. Driving safety and driving efficiency have been enhanced in [158] by incorporating v_d and allowable relative distance into the reward function.

Training efficiency and interpretability have been improved in [159]. Training efficiency is achieved through optimization-embedded DRL for adaptive decision-making, and interpretability is enhanced by transparent model-based optimization. Driving safety and unselfishness have been achieved in [160], with safety ensured by penalizing collisions and ramping off the road, and unselfishness promoted by using MARL to maximize collective benefits. Driving safety and interpretability have been achieved in [161], with safety maintained through penalties for collisions with HDVs and walls. Interpretability is supported by gradual training mode, similar to human learning, where the system starts with sparse traffic and progresses to dense traffic later.

2.5.3 Three-factor Methods for Roundabout Driving

Driving safety, driving efficiency, and training efficiency have been improved in [162]. Driving safety and driving efficiency are promoted through safety and efficiency rewards, respectively. Training efficiency is enhanced via trust region policy optimization (TPRO), which converges faster than PPO and DDPG. In [163], driving safety and driving efficiency have been achieved by rewards for non-collision lane-changing and the difference between initial and target velocities, respectively. Training efficiency is improved by embedding LSTM into the actor-critic network. Training efficiency, driving safety and efficiency have been enhanced in [164]. Training efficiency is improved by normalizing the initial reward for faster convergence. Driving efficiency and driving safety benefit from multiple environments where agents are trained simultaneously, achieving higher success rates and fewer crashes.

2.5.4 Four-factor Methods for Roundabout Driving

Driving safety, driving efficiency, training efficiency, and unselfishness have been addressed in [165], where safety is maintained using d_s, and driving efficiency is enhanced by the ratio of initial to target velocity. Training efficiency is improved through a synthetic representation mechanism that enhances agents' understanding, and unselfishness is promoted using MARL to maximize joint benefits. Driving safety, driving efficiency, interpretability, and training efficiency have been addressed in [65], where safety is ensured via crash penalties and efficiency via high-speed rewards. Interpretability is maintained using the IDM for safe, transparent algorithmic-following. Training efficiency is improved through an interval prediction model to precompute feasible paths, reducing training computation. Driving safety, driving efficiency, training efficiency, and interpretability have been enhanced in [166]. Safety and efficiency are promoted through penalties for collisions and vehicle-stop maneuvers, respectively. Training efficiency is increased by integrating DDPG, DQN, and NMPC. Interpretability is enhanced via the NMPC. Compared with this [166], [165] also addresses four factors by using d_s for safety and the ratio of initial to target velocity for efficiency, with training efficiency improved through a synthetic representation mechanism and unselfishness promoted using MARL; however, it lacks an

explicit interpretability mechanism akin to the NMPC in [166]. Similarly, [65] ensures safety via crash penalties and driving efficiency via high-speed rewards, while employing the IDM to maintain transparency and an interval prediction model to precompute feasible paths for improved training efficiency. Although effective, its approach to interpretability and training efficiency does not reach the level of integration observed in [166]. In the case of three-factor methods, [162], [163], and [164] focus on safety, efficiency, and training efficiency without explicitly addressing interpretability. Their mechanisms, such as using trust region policy optimization, embedding LSTM into actor-critic networks, or normalizing initial rewards and training in multiple environments, provide improvements in training convergence and driving performance, but they do not offer the explicit, interpretable decision-making structure that NMPC affords in [166]. Collectively, while each approach exhibits its own merits, the method in [166] stands out for its balanced integration of all four factors, thereby serving as a promising baseline for future research in roundabout driving.

2.5.5 Five-factor Methods for Roundabout Driving

All five factors have been considered in [106], where driving safety and interpretability are ensured by a rule-based action inspector. Driving efficiency is enhanced via high-speed rewards. Training efficiency is achieved through a Kolmogorov-Arnold network-enhanced DQN. Unselfishness is promoted through rule-based route planning that considers the varying distributions of HDVs on the roundabout. The DRL-based decision making in roundabouts based on DDTUI is summarized in Table 2.3.

Table 2.3
Evaluation of the DRL-based decision making at roundabouts

Ref.	Year	Safety	Efficiency	Training Efficiency	Unselfishness	Interpretability
[153]	2019	-	SAC with higher peak rewards	-	-	-
[154]	2019	-	-	Action repeat, asynchronous advantage	-	-
[155]	2024	-	-	ODD-embedded DQN	-	-
[156]	2020	-	CRM	CRM	-	-
[157]	2022	-	-	Expert guidance	-	Expert guidance
[158]	2024	Allowable relative distance	Desired velocity	-	-	-
[159]	2021	-	-	Optimization-embedded RL	-	Model-based optimization
[160]	2021	Collision penalties	-	-	MARL	-
[161]	2023	Collision penalties	-	-	-	Gradual training
[162]	2023	Safety rewards	Efficiency rewards	TPRO	-	-
[163]	2024	Non-collision rewards	Velocity difference rewards	LSTM-embedded actor-critic	-	-
[164]	2020	Fewer crashes	Higher success rates	Reward normalization	-	-
[165]	2020	Safety distance	Velocity ratio	Synthetic representation	MARL	-
[65]	2016	Crash penalties	High-speed rewards	Interval prediction	-	IDM
[166]	2023	Collision penalties	Vehicle-stop penalties	DDPG, DQN, NMPC integration	-	NMPC
[106]	2024	Rule-based inspector	High-speed rewards	KAN-DQN	Rule-based planning	Rule-based inspector

2.6 Deep Reinforcement Learning-based decision-making at Unsignalized Intersections

2.6.1 Single-factor Methods for Intersection Driving

Traffic efficiency has been improved by using the difference between v_0 and v_d as a reward in [167]. Additionally, a penalty is applied when the velocity drops below a threshold, further boosting traffic efficiency. In [168], driving efficiency has been achieved by applying a constant penalty as long as the AV has not reached the target exits.

2.6.2 Dual-factor Methods for Unsignalized Intersection Driving

Both driving and training efficiency have been improved in [169], where the driving efficiency is enhanced by using total waiting time (TWT) as part of the reward. Training efficiency is increased by employing a background removal ResNet as the Q-network, resulting in lower TWT than baseline algorithms. In [170], driving efficiency and interpretability have been enhanced. The driving efficiency is improved by using the difference between the v_d and v_0 as part of the reward, while the interpretability is achieved through the use of IDM for safe and transparent vehicle following. Similarly, in [171], both driving efficiency and interpretability have been improved. The former is enhanced by incorporating a velocity-based reward, and the latter is enhanced by applying a safety-based rule policy. In [172], driving efficiency is increased by using the safe distance as a reward and the risky distance as a penalty, resulting in higher success rates. Interpretability is achieved using a model-based transparent method combined with twin delayed deep deterministic policy gradient (TD3). In [173], driving efficiency is enhanced by the ratio of v_0 to $v_{\max}$ as part of the reward, and interpretability is improved by gridding the coordination zone into different granularities, converting risky areas into a matrix format.

Both driving efficiency and training efficiency have been improved in [174]. Driving efficiency is achieved by rewarding goal attainment, and training efficiency is increased by using DQN with common and specific sub-tasks. The common sub-task enables knowledge sharing across tasks, while the specific sub-task helps the system better understand a task's main goal. Training efficiency and unselfishness have been improved in [175] through an incentive communication-assisted MARL. Agents create custom messages to influence other agents' policies, improving coordination and achieving globally optimal decisions. The unselfishness is realized by using MARL to maximize overall profits. In [176], driving efficiency and training efficiency have been improved by the adaptive dual-objective transit signal priority (D2-TSP) algorithm with DDQN. D2-TSP optimizes bus speed, saving time for both passengers and those waiting at downstream stops. Similarly, in [177], cooperative intersection management-enhanced DQN boosts both driving and training efficiency by leveraging connectivity between vehicles.

2.6.3 Three-factor Methods for Unsignalized Intersection Driving

In [178], driving efficiency, unselfishness, and training efficiency have been addressed. Driving efficiency is enhanced by penalizing each low-speed state, while unselfishness is achieved through MARL for maximizing overall profits. Training efficiency is improved with multi-agent DQN, which offers faster convergence than baseline algorithms. In [179], driving efficiency, training efficiency, and interpretability have been integrated. Driving efficiency is increased by rewarding each high-velocity state, and training efficiency is achieved by combining deep Q-learning with transfer learning. Interpretability is improved by using the IDM for safe vehicle following.

In [180], driving safety, driving efficiency, and training efficiency have been incorporated. Driving safety is promoted through collision penalties, and driving efficiency is enhanced by rewarding velocities higher than a baseline. Training efficiency is improved by using a spatial and temporal attention module with SAC. The approach in [180] to promoting driving safety and efficiency through collision penalties and velocity-based rewards is based on the referenced study, which primarily focuses on standard driving scenarios. However, it is acknowledged that this method does not explicitly address edge cases, where RL may exploit weaknesses in the reward function.

To improve DRL performance in edge cases, several techniques can be employed. One effective solution is integrating rule-based conditions to handle safety-critical scenarios beyond standard training. Additionally, applying strict safety constraints during both training and deployment can prevent RL from exploiting reward function loopholes. Techniques such as shielded reinforcement learning, which combines DRL with formal safety verification, further ensure robust decision-making under extreme conditions. While addressing edge cases was not the primary focus of the referenced work, these methods present promising directions for enhancing DRL-based decision-making in future research.

In [181], all three aspects have also been addressed. Driving safety and efficiency are improved by rewarding goal attainment and penalizing collisions. Training efficiency is enhanced using a randomized prior function for each ensemble member, leading to a better Bayesian posterior [182]. Compared with this approach, [178] focuses on driving efficiency by penalizing each low-speed state and promotes unselfishness through multi-agent reinforcement learning, with training efficiency improved via multi-agent DQN; however, its safety mechanism is less explicitly defined and does not incorporate the goal-oriented reward strategy used in [181]. Similarly, [179] increases driving efficiency by rewarding each high-velocity state and enhances training efficiency by combining deep Q-learning with transfer learning, while interpretability is achieved by employing the IDM for safe vehicle following. Although this offers a more transparent decision-making process, it lacks the robust uncertainty estimation provided by the randomized prior function in [181]. Furthermore, [180] promotes driving safety and efficiency through collision penalties and velocity-based rewards, and improves training efficiency using a spatial and temporal attention module with SAC. While its approach is similar in spirit to that of [181], it primarily focuses on standard driving scenarios and does not explicitly address edge cases where reinforcement learning might exploit reward function loopholes. Collectively, these comparisons highlight that while each method exhibits unique strengths, the approach in [181] offers a more balanced integration of safety, efficiency, and training efficiency, particularly through its use of randomized prior function for robust uncertainty handling.

2.6.4 Four-factor Methods for Unsignalized Intersection Driving

In [183], driving safety, driving efficiency, training efficiency, and unselfishness have been incorporated. Driving safety is enhanced through autonomous intersection management (AIM), and driving efficiency is improved by applying a constant penalty until the AV reaches the exits. Training efficiency is improved by embedding AIM and LSTM into the learning, and unselfishness is achieved through MARL. The framework in [183], based on SUMO simulation, enhances training efficiency by integrating the AIM and LSTM networks. AIM prioritizes critical interactions, while LSTM captures historical dependencies, improving optimization. However, onboard training remains impractical due to computational complexities, with training typically performed offline for real-time deployment.

In [184], driving safety, driving efficiency, training efficiency, and interpretability have been integrated. Driving safety is promoted through collision penalties, and driving efficiency is enhanced by rewarding goal attainment. Training efficiency is improved through the Mix-Attention Network, synthetic representation mechanism, and replay memory mechanism. The interpretability is ensured by using the IDM.

Table 2.4
Evaluation of the DRL-based decision making at unsignalized intersections

Ref.	Year	Safety	Efficiency	Training Efficiency	Unselfishness	Interpretability
[167]	2021	-	Velocity difference reward	-	-	-
[168]	2020	-	Time penalty	-	-	-
[169]	2022	-	Total waiting time	Background removal ResNet	-	-
[170]	2020	-	Velocity difference reward	-	-	IDM
[171]	2022	-	Velocity-based reward	-	-	Safety-based rule policy
[172]	2022	-	Safe distance reward	-	-	MPC with TD3
[173]	2023	-	Velocity ratio reward	-	-	Gridded coordination zone
[174]	2020	-	Goal attainment reward	DQN with subtasks	-	-
[175]	2024	-	-	Incentive communication	MARL	-
[176]	2023	-	D2-TSP	DDQN	-	-
[177]	2023	-	CIM-enhanced DQN	CIM-enhanced DQN	-	-
[178]	2020	-	Low-speed penalty	Multi-agent DQN	MARL	-
[179]	2021	-	High-velocity reward	DQL with transfer learning	-	IDM
[180]	2021	Collision penalties	High-velocity reward	SAC with attention	-	-
[181]	2020	Collision penalties	Goal attainment reward	RPF	-	-
[183]	2022	AIM	Constant time penalty	AIM and LSTM	MARL	-
[184]	2024	Collision penalties	Goal attainment reward	Mix-Attention Network	-	IDM
[185]	2023	Collision penalties	Low-velocity penalty	VD-MADQL	MARL	IDM

2.6.5 Five-factor Methods for Intersection Driving

In [185], driving safety has been promoted through collision penalties, while driving efficiency is enhanced by penalizing each state with velocity lower than the $v_{\min}$. Training efficiency is improved using value decomposition-based multi-agent deep Q-learning. Unselfishness is achieved by employing MARL to minimize joint profits, and interpretability is ensured through the IDM. The DRL-based decision making on unsignalized intersections based on DDTUI is summarized in Table 2.4.

2.6.6 Five-factor Methods for Intersection Driving

Finally, all the five factors are addressed in a few studies. In [185], driving safety has been promoted through collision penalties, while driving efficiency is enhanced by penalizing velocities lower than the predefined $v_{\min}$. Training efficiency is improved using value decomposition-based multi-agent deep Q-learning. Unselfishness is achieved by employing MARL to minimize joint profits, and interpretability is ensured through IDM. The whole evaluation based on DDTUI is summarized in Table 2.4.

2.7 Summary

This chapter presents a comprehensive overview of the current state of the art in DRL-based decision-making for autonomous vehicles. By discussing recent research efforts in this field, this chapter highlights the diverse algorithms developed to address decision-making tasks across various scenarios, including highways, on-ramping merging, roundabouts, and unsignalized intersections. Our analysis goes beyond simply presenting these algorithms by uncovering valuable insights, identifying key gaps in the current research, and highlighting emerging trends in DRL-based decision making for autonomous driving. Current DRL algorithms, such as DQN and PPO, often require additional mechanisms, like reward shaping or attention modules, to comprehensively achieve DDTUI objectives. DQN, as a value-based method, performs effectively in discrete-action environments, enhancing training efficiency and driving safety. PPO, being policy-based, excels in continuous-action settings, promoting smoother driving behavior and unselfishness

through adaptive decision-making. While both algorithms contribute to solving DDTUI challenges, their effectiveness varies depending on the driving context, highlighting the need for adaptation for specific tasks and driving environments. While driving efficiency and safety are addressed across most studies, there is a growing trend towards addressing multiple DDTUI factors concurrently. Emerging approaches, such as MARL and the integration of traditional control methods with DRL, show promise in tackling complex challenges with increased unselfishness and interpretability in autonomous driving.

Based on existing studies, Table V summarizes the distribution of evaluation factors considered in four typical scenarios. Most studies, i.e., 17 studies that account for 94.4% of existing studies, prioritize efficiency at intersections to optimize travel time in complex and interaction-heavy environments. Efficiency is also emphasized at ramps in 14 studies (i.e., 87.5%) to reduce congestion and streamline traffic flow. Safety is particularly emphasized on highways in 23 studies (i.e., 76.7%), which addresses the importance of accident prevention in high-speed settings. In contrast, intersections address safety less often. Training efficiency is significant at roundabouts in 12 studies (i.e., 75%) and unsignalized intersections in 12 studies (i.e., 66.7%). This reflects a need for effective training methods to ensure smooth vehicle maneuvering in these challenging contexts. Interpretability is particularly valued at ramps in 9 studies (i.e., 56.25%) and on highways in 11 studies (i.e., 36.7%), respectively. This emphasizes understandable decision-making in these areas. Unselfishness receives less emphasis overall, although highways and ramps give it much attention. Unselfish driving plays a crucial role in enhancing overall driving performance and ensuring smooth traffic flow. Compared to selfish driving, where vehicles prioritize individual objectives, unselfish driving promotes cooperative behaviors, reducing abrupt lane changes, minimizing congestion, and improving road safety. This cooperative approach facilitates smoother merging and lane-changing, benefiting both individual drivers and the overall traffic system. However, unselfishness is not an absolute requirement in all scenarios. While it is generally advantageous, special conditions, such as emergencies, may necessitate prioritizing self-interest for safety or efficiency. As such situations are relatively rare, it is recommended to maintain an unselfish driving norm under typical conditions while ensuring adaptive mechanisms for handling exceptional scenarios. Future challenges are summarized as

Table 2.5
Occurrence and Ratio of Evaluation Factors Across Different Scenarios

Scenario	Safety	Efficiency	Training Efficiency	Unselfishness	Interpretability
Highway	23 (76.7%)	20 (66.7%)	11 (36.7%)	13 (43.3%)	11 (36.7%)
Ramp	9 (56.25%)	14 (87.5%)	8 (50%)	8 (50%)	9 (56.25%)
Roundabout	10 (62.5%)	11 (68.75%)	12 (75%)	3 (18.75%)	6 (37.5%)
Intersection	5 (27.7%)	17 (94.4%)	12 (66.7%)	4 (22.2%)	6 (33.3%)
Total	47 (58%)	63 (77.8%)	44 (54.3%)	29 (35.8%)	32 (39.5%)

The numbers in parentheses indicate the percentage of the total studies for each factor.

1. Achieving a balance between all five DDTUI factors in a single framework: This chapter reveals that while many studies addressed multiple DDTUI factors, very few managed to incorporate all five factors simultaneously. For instance, only 3 out of 16 studies in roundabout scenarios and 1 out of 19 studies in intersection scenarios addressed all five factors. This highlights the complexity of developing a unified framework that can effectively balance DDTUI. Future research should focus on developing integrated frameworks that can holistically address five DDTUI factors concurrently.
2. Improving the interpretability of DRL models without sacrificing performance: While some studies have made strides in improving interpretability, such as using IDM for interpretable car-following, many high-performing DRL models remain black boxes. Out of the reviewed papers, less than 40% explicitly addressed interpretability. Furthermore, most papers considering interpretability use only one method. In the future, multiple interpretability methods can be applied to enhance interpretability, such as using the APF and IDM concurrently.
3. Enhancing the unselfishness of AVs in complex, multi-agent environments: While approximately 50% of studies use MARL to promote unselfishness, the complexity of real-world traffic scenarios presents uncertainties of driving behaviors. Future research should explore more sophisticated MARL techniques based on real-world experience. For example, combining game theory with driving style classification based on real-world datasets can better model the behaviors of HDVs.

Chapter 3

Balanced Exploration and Attention-inspired Decision Making on Highways

Autonomous driving has attracted great interest due to its potential capability in enhancing safety and improving traffic efficiency. Both model-based and learning-based methods are widely used in autonomous driving. Out of which, model-based methods rely on existing events in the dataset but are poor in learning extended situations [153]. As a comparison, the Deep Q-Network (DQN) has a strong capability in learning within interactive driving. However, existing DQN faces challenges in convergence in terms of speed and accuracy, especially in interactive environments. Furthermore, the poor convergence causes high risks of collisions and slow driving speed. Therefore, this chapter presents a modified DQN to achieve a lower number of collisions, higher average driving speed, and faster convergence during interactive driving. Besides, interpretability and unselfishness are also considered in this chapter to satisfy the DDTUI discussed in the previous chapter. The modified DQN is developed by introducing a risk-attention mechanism, a balanced reward function, and a collision-supervised mechanism (RBDQN-CS). The risk-attention mechanism enhances the DQN to pay attention to high-frequent interactions. The balanced reward function specifies the weight of the control strategy to handle the interactions with surrounding human driven vehicles. The collision-supervised mechanism detects the collision risks and prevents the collision occurrence during lane-changing. Simulation results demonstrate that the proposed RBDQN-CS outperforms DQN and other popular baseline DRL algorithms.

The selected KPIs, such as collision rate, average driving speed, and convergence speed, are widely recognized in current DRL-based autonomous driving research for evaluating driving performance, as discussed in Chapter 2. These KPIs align with the DDTUI framework, which is among the most commonly accepted evaluation frameworks in recent studies. While other KPIs could also provide valuable insights for real-world driving, this study focuses on DDTUI as the primary evaluation method. Future work could further expand the evaluation scope by incorporating additional performance indicators.

3.1 Introduction

AVs faces challenges in making reliable decisions when interacting with HDVs. This is attributed to the difficulty in accurately predicting intentions of HDVs, especially on highways. In the real-world setting, HDVs driving on highways may frequently perform lane-keeping, lane-changing, acceleration, and deceleration. To predict the intentions of HDVs, some model-based methods have been developed [186]. However, these model-based methods struggle to adapt well to complex traffic environments, as the interactions with HDVs are typically oversimplified. Model-based methods typically assume regular movements and constant speeds for surrounding HDVs, simplifying interactions to reduce computational complexity. However, such assumptions limit the variability of HDV behavior, resulting in low unpredictability and reduced robustness when facing real-world driving conditions, where driver behavior is dynamic and less predictable. In contrast, DRL-based approaches can better account for HDV unpredictability by introducing randomized behaviors during training. By simulating diverse HDV actions, such as varying speeds, unexpected lane changes, and irregular driving patterns, DRL enables the agent to adapt its decision-making strategies in real-time. Learning-based methods provide alternative solutions to enable the agent to explore complex situations thoroughly, thereby making them more suitable for interactive driving. For example, DRL can generate the optimal control strategy to handle the complex traffic environment [187]. This is because DRL enables the AV to fully interact with HDVs during training, filtering out strategies that fail to ensure safety and efficiency.

Existing DRL algorithms, such as a DQN [188], perform well in short-term driving scenarios. However, these algorithms still have limitations in the learning during the long-term driving. First, it is difficult for DQN algorithms to obtain the global optimal solution in long sequences because they struggle with long-term dependencies, which may cause collisions. Second, DQN algorithms treat the non-interactive driving and interactive driving of the AV as the same cases. Therefore, the performance of the DQN might be compromised in interactive driving. Third, collision detection is not considered in DQN algorithms during the lane-changing process, which may cause substantial economic loss and pose severe safety risk. Moreover, the lack of a collision detection phase makes the DQN less interpretable in collision-avoidance.

To address the aforementioned limitations, this chapter proposes a modified DQN framework by introducing the risk-attention mechanism, the balanced reward function, and the collision-supervised mechanism (RBDQN-CS). The risk-attention mechanism quantifies the risk levels around surrounding HDVs, thereby enabling AVs to focus on interaction areas, increasing training efficiency. The balanced reward function facilitates balanced exploration from a global perspective. In the balanced reward function, the weight of rewards increases when the interactions occur frequently. Consequently, the updated control strategies are derived mainly from interaction areas. The collision-supervised mechanism uses a double-side collision detection, aiming to filter out certain predicted and detected collisions, enabling the agent to explore much complex uncertainties and thereby increasing the training efficiency. Namely, collision risks from both the front vehicle (FV) and the rear vehicle (RV) during lane-changing are considered. In addition, the DRL is combined with MPC as a safe and generalizable control strategy [189]. The main contributions include

- An RBDQN-CS is proposed that achieves lower collision rates and higher average speeds, as well as higher rewards and faster convergence from a learning perspective, compared to benchmark algorithms [34, 38, 190, 191].
- The collision rate when driving on a highway is significantly reduced by using the proposed collision-supervision mechanism. Furthermore, this mechanism enhances the interpretability of the DQN [190] in interactive driving.

- From a traffic perspective, combining DQN with either a risk-attention mechanism or a balanced reward function individually results in lower collision rates, higher average speeds, faster convergence, and higher rewards compared to DQN [190].

3.2 Related Works

State-of-the-art results of using DRL have been demonstrated in AVs [192], including Q-Learning [30], deep deterministic policy gradient (DDPG) [193], proximal policy optimization (PPO) [194], and DQN [190].

In Q-Learning, the state-action value function is utilized to determine the best action in a given state [30]. For example, correct actions are selected by Q-Learning in autonomous driving despite numerous safety constraints [31]. However, this method can only be used in simple tasks and scenarios [31]. Moreover, The convergence of Q-Learning is slow [32].

DDPG uses deep neural networks to approximate the control policy [193]. This method has been demonstrated to effectively improve the convergence and driving performance [38,39]. However, this method has limitations in the exploration of more possible actions and the adaptability in diverse driving scenarios because it obtains an absolute result from the control policy. The absolute result restricts the algorithm's flexibility, as it discourages exploration of alternative actions that could be more suitable.

PPO utilizes the control policy in a probability distribution and offers faster exploration improvements over DDPG [195]. PPO has been used to create control models for multi-agent driving scenarios and crowded highway traffic [33,34]. In PPO, the reward function is often connected to the general performance instead of emphasizing the performance of required objective. Therefore, the convergence efficiency of PPO for long training sequences is low, as many irrelevant sections dilute the performance of crucial sections.

DQN has been widely utilized in various traffic simulations and is well-suited for tasks like navigation during the interactive driving and intersection management [196]. Its discrete-action framework is applicable to driving tasks with discrete commands, such as accelerating, decelerating, lane-changing, and lane-keeping. Therefore, DQN does not need extra

action discretization in driving tasks. In addition, DQN uses experience replay to learn from past experiences more efficiently. Moreover, DQN is generally less computationally intensive, which is critical in real-time driving [35]. However, DQN is not suitable for complex driving scenarios because it struggles with long-term dependencies, leading to suboptimal decision-making in complex interactive environments. Moreover, a component that provides interpretability for the collision-avoidance in driving is needed.

In summary, existing DRL algorithms face challenges in discrete-action frameworks, high computational load, and low convergence speed, relying on the extra information of environment. Despite DQN addresses some of these challenges by employing discrete-action framework, its convergence slows down in long-term dependencies, such as long-term driving on highways. In addition, the inherent risk of collisions during interactions with HDVs remains unresolved due to the same attention given to input features and low interpretability. Furthermore, it is essential for balanced exploration in long-term driving to construct a suitable probability distribution for DQN. To address these issues, the risk-attention mechanism, the balanced reward function and the collision-supervised mechanism are needed to be introduced into DQN.

3.3 Problem Formulation and the Decision-making Framework

3.3.1 Problem Formulation

Fig. 3.1 depicts the considered interactive driving scenario and three cases. Fig. 3.1(a) shows the considered interactive driving scenario. Fig. 3.1(b) to Fig. 3.1(d) present three cases. The upper sub-figures of Fig. 3.1(b) to Fig. 3.1(d) depict normal autonomous driving mode. The lower sub-figures of Fig. 3.1(b) to Fig. 3.1(d) illustrate the desired mode, focusing on safety, efficiency, and unselfishness.

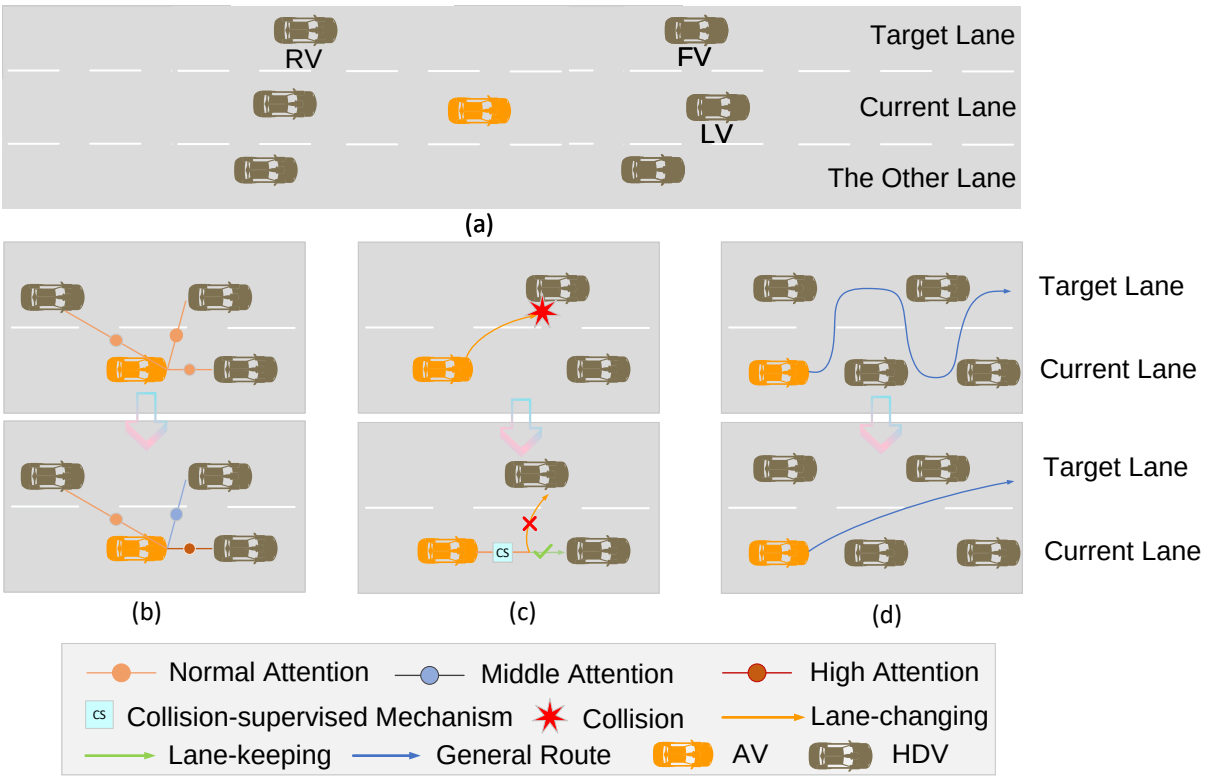


Figure 3.1: Performance of different driving. (a) the interactive driving on a highway; (b) attention-based interactive driving; (c) collision-supervised interactive driving; (d) unselfish interactive driving.

Fig. 3.1(a) presents a scenario that involves in a three-lane highway: the target lane, the current lane, and the other lane. In this scenario, the AV follows the lead vehicle (LV) in current lane. The AV would interact with the front vehicle (FV) and rear vehicle (RV) if it changes to the target lane, and interact with other HDVs if it changes to the other lane. Since the lane-changing process is same, this study focuses on the lane-changing maneuver from the current lane to the target lane. To describe the scenario clearly, some statement are presented as follows.

Denote v_{AV}, v_{FV}, and v_{RV} as the speed of the AV, FV, and RV, respectively; d_{FV} and d_{RV} are the longitudinal distance between the AV and the FV and between the AV and the RV, respectively. $p_{\mathrm{AV}}(t)$, $v_{\mathrm{AV}}(t)$, and $h_{\mathrm{AV}}(t)$ denote the position, speed, and heading of the AV at time t, and $p^i_{\mathrm{SV}}(t)$, $v^i_{\mathrm{SV}}(t)$, and $h^i_{\mathrm{SV}}(t)$ represent the position, speed, and heading of the i-th surrounding HDV at time t. $v_{\min}$ and $v_{\max}$ denote the minimum and maximum allowable speed for the AV, respectively. $u_{\min}$ and $u_{\max}$ denote the minimum and maximum allowable acceleration for the AV, respectively. $\delta_{\min}$ and $\delta_{\max}$ denote the

minimum and maximum allowable steering angle for the AV, respectively. $u(t)$ and $\delta(t)$ denote the acceleration and the steering angle for the AV at timestep t, respectively. L is the wheelbase length. $u(t)$ is the acceleration of the AV at timestep t. $\delta(t)$ is the steering angle of the AV at timestep t. d_s represents predefined safe distance.

HDVs are assumed to exhibit random driving behaviors, enhancing the realism and fidelity of the simulations. During driving, three aspects are important for the AV: safe decision-making, efficient path planning, and unselfish driving. Safe decision-making enables the AV to judge whether each HDV surrounded with high risks, avoiding interaction with high-risk HDVs. Efficient path planning allows the AV to reach the target position faster, reducing time loss. Unselfish driving minimizes unnecessary interactions with HDVs, reducing the complexity of interactive driving.

Fig. 3.1(b) compares driving performance of the AV with and without using an attention mechanism. In the upper sub-figure, the AV assigns equal importance to all surrounding HDVs when an attention mechanism is not used. In the lower sub-figure, the AV deferentially prioritizes three nearby HDVs across three levels when an attention mechanism is used. This different priority setting helps the AV better understand potential risks and adapt to the current traffic environment. Fig. 3.1(c) compares driving performance of the AV with and without using a collision-supervised mechanism. The upper sub-figure depicts that without using a collision-supervised mechanism, the AV collides with a HDV in the target lane because it would change lane. From the lower sub-figure, when a collision-supervised mechanism is used, the AV would keep the lane to avoid the collision. Fig. 3.1(d) illustrates driving performance of the AV with and without using an unselfish driving mechanism. In the upper sub-figure, the AV changes the lane which disturbs all nearby HDVs without using an unselfish driving mechanism. Collisions can be avoided in such a situation, however the complexity of the AV's decision-making increases as the frequency of interaction between AVs and HDVs increases. The lower sub-figure demonstrates that the AV performs a simple and appropriate lane-changing maneuver when an unselfish driving mechanism is used. This situation affect only two surround-

ing HDVs and prevents other HDVs from unnecessary interactions. Based on the above analysis, it can be found that attention-based, collision-supervised, and unselfish driving mechanisms can effectively enhance driving performance of AVs from the perspective of the safety and rationality when interacting with HDVs on highways.

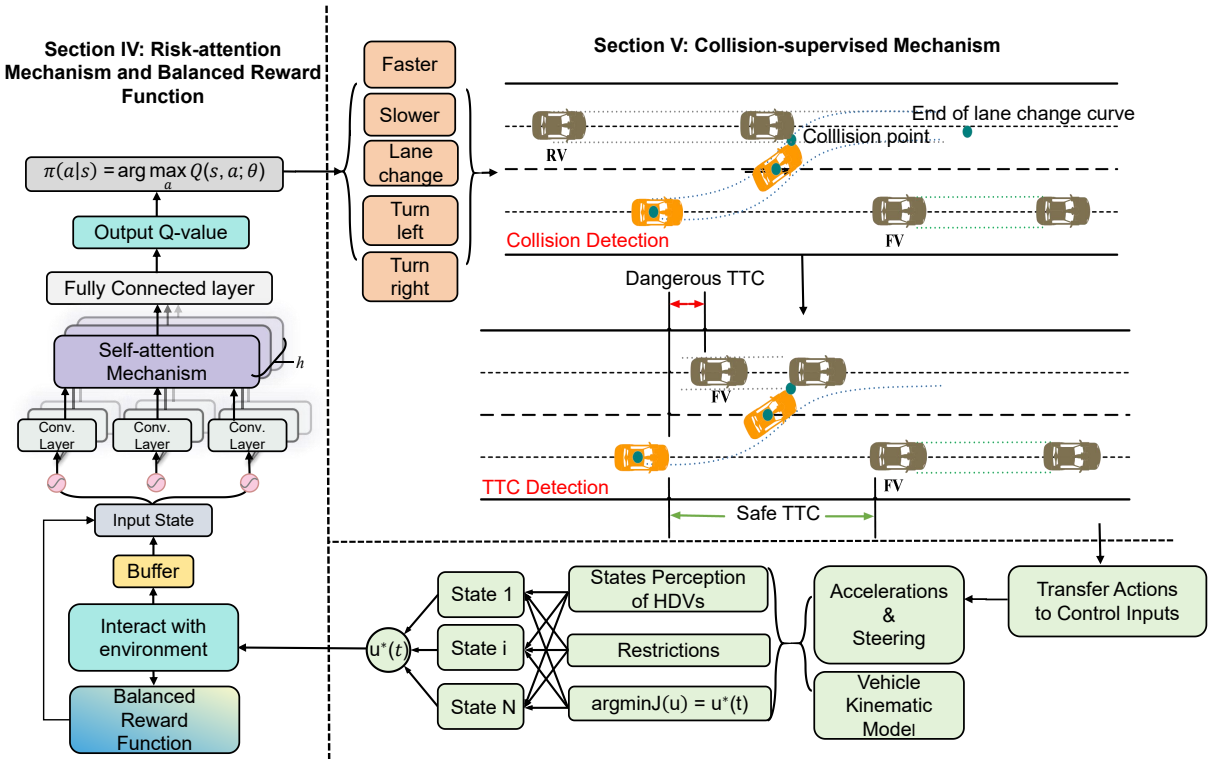


Figure 3.2: The decision-making framework.

3.3.2 The Decision-making Framework

The interactive driving with safety, efficiency and unselfishness on a three-lane highway is achieved by the proposed RBDQN-CS framework, as illustrated in Fig. 3.2. In this framework, the three mechanisms are categorized into two critical aspects, corresponding to two sections of this chapter: the risk-attention mechanism and balanced reward function, and the collision-supervised mechanism. Attention to interaction areas is achieved through the proposed risk-attention mechanism. The balanced update of the control strategy for interaction areas is realized via the proposed balanced reward function. The collision-supervised mechanism uses collision detection and time-to-collision (TTC) detection to avoid detectable collisions with RV and FV, respectively. The collision-supervised mech-

anism addresses the shortcomings identified in previous DQN work, where the collision detection phase was missing [190]. In addition, MPC is implemented to translate planned actions into safe and smooth control commands (accelerations and steers), enhancing safety and robustness throughout the driving.

3.4 Risk-attention Mechanism and Balanced Reward Function

The risk-attention mechanism is embedded with the network structure of the DQN framework to emphasize the interaction areas. The balanced reward function is integrated with the Q-function of the DQN framework to efficiently update control strategies during interactive driving.

3.4.1 Network Structure of the Risk-attention Mechanism

Fig. 3.3 illustrates the network structure of the risk-attention mechanism, which includes the input layer, convolutional neural network layer, risk-attention layer, fully connected layer, and output layer.

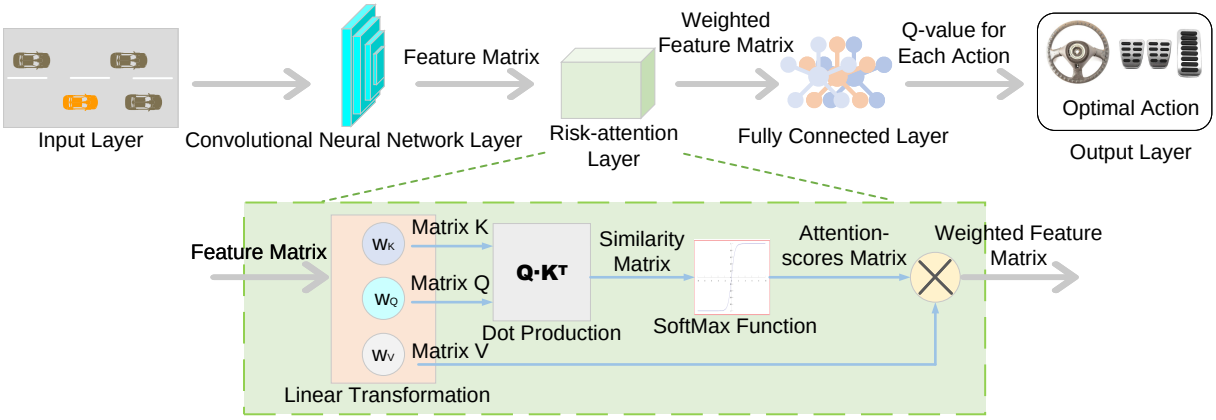


Figure 3.3: Network structure of the risk-attention mechanism.

3.4.1.1 Input Layer

The input layer receives raw images from the vehicle's onboard sensors, encapsulating the AV's current state, and is represented by the state vector s.

3.4.1.2 Convolutional Neural Network Layer

The raw data is transformed by convolutional neural network (CNN) layers into a feature set F that helps pattern recognition and decision-making:

$$F = CNN(s; \theta_{\text{CNN}}) \tag{3.1}$$

where θ_{CNN} are the parameters of the CNN.

3.4.1.3 Risk-attention Layer

Inspired by [197], a risk-attention layer is designed in this chapter. The risk-attention layer processes features extracted by the convolutional neural network layers while focusing on critical features that influence decision-making during navigation. The input to this attention layer is the feature matrix F, obtained from the CNN layers, where each row of F represents features derived from different parts of the input image. The risk-attention layer first calculates attention scores, which determine the importance priority of each feature when compiling the final output. This is achieved using a set of trainable weights that is often structured as three matrices known as Query (Q), Key (K), and Value (V) matrices. To obtain the Q, K, and V matrices from the feature matrix F, learned linear transformations are used. Each of these matrices is derived from F using separate linear transformations. Denote the linear transformations as W_Q, W_K, and W_V respectively. The transformations are applied to F as

$$Q = F \cdot W_Q, \quad K = F \cdot W_K, \quad V = F \cdot W_V \tag{3.2}$$

The attention score for each feature is computed using the dot product of Q and K:

$$A = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right) \tag{3.3}$$

where d_k represents the dimensionality of the K, which normalizes the dot products to prevent them from growing too large. The attention scores matrix A is then used to create a weighted sum of the value vectors, which forms the output of the attention mechanism:

$$F' = A * V \tag{3.4}$$

This step effectively allows the network to focus on the most relevant features. The output F' then feeds into the fully connected layers to calculate the Q-values. The risk-attention layer allows the network to dynamically adjust itself by focusing on important features. These important features are dependable on the current driving context on highways. During the converging, the parameters θ_{att} of the risk-attention layer (comprising the weights of the Q, K, and V matrices) are updated to optimize the attention given to the most important features.

3.4.1.4 Fully Connected Layer

The fully connected layer approximates the Q-function by mapping the processed features F' to Q-values for candidate actions a':

$$Q(s, a'; \theta) = f(F'; \theta_{\text{fc}}) \tag{3.5}$$

where θ_{fc} indicates the parameters of the fully connected layers. θ represent the weights of all neural networks in risk-attention mechanism used to approximate the Q-value function.

3.4.1.5 Output Layer

The output layer provides the estimated Q-values for each possible action, forming the basis for the optimal decision-making:

$$a^* = \arg\max_{a'} Q(s, a'; \theta) \tag{3.6}$$

where a^* is the optimal action selected by the network.

3.4.2 Network Policy of the Risk-attention Mechanism

The network policy governs the AV's behavior by determining actions based on the Q-values. During the converging of DQN, the determination of new candidate actions is crucial for policy exploration. The risk-attention layer enhances policy exploration by extracting key features from the feature matrix. In addition, exploitation enables the risk-attention mechanism to select optimal actions. Ultimately, a converged policy with safety and efficiency is obtained.

3.4.2.1 Action Determination

In action determination, an epsilon-greedy policy is employed to strike a crucial equilibrium between exploration of new actions a_t at timestep t and known control strategies:

$$a_t = \begin{cases} \text{random action with probability } \varepsilon, \\ \arg\max_a Q(s, a; \theta) \text{ with probability } 1 - \varepsilon. \end{cases} \tag{3.7}$$

This policy facilitates the discovery of new control strategies and navigates the highway more efficiently.

3.4.2.2 Adaptive Exploration Rate

The value of ε is adaptively decayed, allowing the AVs to transit from an exploratory phase to a more exploitation-focused phase. The exploitation-focused approach enables the AV to significantly rely on the past learning. The AV uses the optimal control strategy derived from past learning to make decisions efficiently and safely. The value of ε is employed to balance exploration and exploitation during training. It starts at $\varepsilon = 0.9$, encouraging exploration in the early stages. The value decays by 0.15 every 1000 episodes until reaching a minimum of $\varepsilon = 0.1$, ensuring a gradual transition toward exploitation while retaining some exploratory behavior to avoid premature convergence to suboptimal policies. As ε updates, the AV transits from exploring random actions to exploiting optimal actions, leading to more reliable driving behaviors. Therefore, ε handles the complexities of interactive driving.

3.4.2.3 Integration with the Risk-attention Layer

The risk-attention layer enhances the policy exploration by ensuring that the AV pays more attention to the most salient features of HDVs. These features, including the proximity and speed of surrounding HDVs, are critical when making decisions in a congested environment on highways.

3.4.2.4 Exploration vs. Exploitation

Exploration involves in testing various maneuvers, including lane-keeping,lane change, and speed adjustments, to optimize the AV's path on highways. Exploitation take advantages of the accumulated experience of the AV's to select optimal actions with safety, efficiency and unselfishness.

Algorithm 1: RDQN Learning Algorithm

Input: Replay memory capacity N, total episodes M, total timesteps T, update frequency C, and exploration probability ε
Output: Learned action-value function Q
Initialize replay memory $\mathscr{D}$ to capacity N
Initialize action-value function Q with random weights θ
Initialize target action-value function $\hat{Q}$ with weights $\theta \leftarrow \theta$
for episode $= 1$ **to** M **do**
  Initialize sequence s_1 and preprocessed sequence ϕ_1
  for timestep $t = 1$ **to** T **do**
    if with probability ε **then**
      Select a random action a_t
    else
      Select $a_t \leftarrow \arg\max_a Q(\phi(s_{t+1}), a'; \theta)$
    end if
    Execute action a_t by attention-Q-network and observe reward r_t and new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathscr{D}$
    Sample minibatch from $\mathscr{D}$
    Set target y_j based on Bellman equation
    Compute the predicted Q-value $\hat{y}_j \leftarrow Q(\phi(s_{t+1}), a'; \theta)$
    Calculate *Loss* using Mean Squared Error
    Perform a gradient descent step by *Loss*
    if every C steps **then**
      Reset $\hat{Q} \leftarrow Q$
    end if
  end for
end for

3.4.2.5 Convergence to the Optimal Control Strategy

The optimal control strategy is expected to be reached based on the accumulated driving experience of AVs. The optimal control strategy prioritizes the safety and efficiency of AVs by minimizing both collisions and travel time.

3.4.3 Learning of Risk Attention-assisted DQN (RDQN)

The learning algorithm updates the weights of the network to optimize the policy. The updating leverages the risk-attention mechanism and reward function. As shown in Algorithm 1, the replay memory, action-value function, and target action-value function are initialized. The replay memory stores the experiences $(state, action, reward, next\,state)$ in a replay buffer during the converging. The replay buffer makes full use of historical data.

The network periodically samples a batch of experiences from this buffer for learning purpose. Then for timestep t, the target Q-value y_j is updated by the Bellman equation. The calculating process is formulated as

$$y_j = r + \gamma \cdot \max_{a'} Q(\phi(s_{t+1}), a'; \theta) \tag{3.8}$$

where r is the current reward, γ is the discount factor, s_{t+1} is the state at timestep $t+1$ resulting from taking action a_t in state s_t, and ϕ is a function that transforms the raw state input into a feature representation. $\max_a Q(\phi(s_{t+1}), a'; \theta)$ is the maximum Q-value for s_{t+1} over a', predicted by the target network with θ.

To stabilize learning, the network maintains a separate and slowly updated target network. The target network provides a fixed target Q-value for updating, which helps stabilize the converging process. The Q-network optimizes the weights by minimizing the difference between the predicted Q-values $\hat{y}_j$ and y_j:

$$Loss = (y_j - \hat{y}_j)^2 \tag{3.9}$$

Once the *Loss* is obtained, the Q-network's weights are updated in the opposite direction of the gradient to minimize the loss. Finally, the parameters of the risk-attention layer are wholly updated and enhanced to better handle complex interactions on highways.

3.4.4 Balanced Reward Function

To formulate the driving problem on highways as a Markov Decision Process (MDP), the key components are the state space, action space and reward function. The state space includes the AV's position, velocity, and heading, as well as the relative positions, velocities, and headings of surrounding vehicles within a specified range. The state at time t is represented as

$$s_t = [p_{\mathrm{AV}}(t), v_{\mathrm{AV}}(t), h_{\mathrm{AV}}(t), p^i_{\mathrm{SV}}(t), v^i_{\mathrm{SV}}(t), h^i_{\mathrm{SV}}(t)]^\top \tag{3.10}$$

The action space is discrete and includes five discrete-actions: accelerate, decelerate, maintain speed, turn right, and turn left. These discrete-actions are subsequently converted into control commands (acceleration and steering) by the MPC controller.

The reward function, called the feedback module, can assess the actions determined by the AV. The velocity and capability of collision avoidance are used to measure the efficiency and safety of the AV, respectively. Specifically, the risk-attention mechanism can help the AV avoid collisions with surrounding HDVs while driving at a highly average speed when an appropriate reward function is provided. However, the traditional reward functions cannot be applied directly because they fail to reflect the interactive process for two reasons. First, the reward in the function is significantly influenced by the previous actions. Therefore, it is impossible to balance the historical and current control strategies in the interactive-orientated tasks when the interaction is not considered in the traditional reward function. Second, most collisions occur to AVs when interacting with HDVs because of high speeds [198]. In such situations, more attention should be paid to the interactive steps. This goal can not be achieved in the traditional reward function as their weights updates without considering the interaction between the AV and HDVs. Third, the longer the sequence length, the less attention is paid to to the performance of interactive steps.

To address the aforementioned issue, a hyperparameter is introduced to balance the historical reward and the reward obtained from the interactive-oriented tasks. Then, a balanced reward function is proposed to update based on the number of interactive HDVs, and Considering the number of surrounding HDVs, the balanced discount factor γ_b is considered as

$$\gamma_b = \gamma + \alpha \cdot \frac{N_{\text{HDVs}}}{\text{max\_HDVs}} \tag{3.11}$$

where N_{HDVs} is the number of surrounding HDVs. max_HDVs is the maximum number of surrounding HDVs the AV has encountered. The balanced value y_b used for updating the Q-value in this adjusted setting is computed as

$$y_b = r + \gamma_b \cdot \max_{a'} Q(\phi(s_{t+1}), a'; \theta) \tag{3.12}$$

If there is a highly interactive process, the current decision becomes more valuable, leading to the control strategies' updating assign a greater value to interactive actions from the current timestep. Therefore, γ_b promotes a safety-aware and forward-looking control strategy during the interaction. The r is formulated as

$$r = r_e + r_s + r_u + r_c \tag{3.13}$$

where r_e is the reward of efficiency, r_s is the reward of safety, r_u is the reward of unselfishness, and r_c is a constant reward. The sub-rewards were selected to balance driving objectives: efficiency, safety, and unselfishness. Fine-tuning was conducted based on a priority-driven approach, where safety was considered the most important factor, followed by efficiency and unselfishness. During training, if the observed driving behavior did not align with this priority order, the corresponding reward values were adjusted by increasing or decreasing them by 25% to 50%, ensuring the final policy reflected the desired balance among the driving objectives. The reward of efficiency is presented to meet the requirements from the perspective from efficiency:

$$r_e = N \quad \text{if the final point is not reached} \tag{3.14}$$

where N is a constant and negative value, representing that there is a fixed penalty for the AV as long as it does not reach the final point. The reward of safety is presented to meet the requirements from the perspective from safety:

$$r_s = M \quad \text{if there is a collision} \tag{3.15}$$

where M is a largely constant and negative value, representing there is a large penalty for AV as long as it encounters collisions. Therefore, the AV is guided to take actions that can avoid any collisions. The reward of unselfishness is presented to meet the requirements from the perspective from unselfishness:

$$r_u = H \quad \text{if only one lane-changing to the target lane} \tag{3.16}$$

where H is a constant and positive value, representing that there is a reward for AV as long as it makes only one or two lane-changing behaviors to the target lane based on its initial position during a long time span. r_u can lead to a smaller number of lane changes, guiding the AV to avoid collisions.

3.5 Collision-supervised Mechanism

This section introduces the collision-supervised mechanism that helps the AVs avoid collisions with surrounding HDVs during the interaction on highway. This mechanism mainly includes three parts: driving rules of HDVs, lane-changing rules of AV, and safety evaluator.

3.5.1 Driving Rules of HDVs

Lane-changing and lane-keeping rules are included.

3.5.1.1 Lane-changing Rules

According the driving rules of HDVs on highways, an HDV has to keep a safe distance from other HDVs ahead. Namely, $d_c > d_s$. Once $d_c < d_s$ is encountered, the HDV changes lanes.

3.5.1.2 Lane-keeping Rules

The Intelligent Driver Model (IDM) [199] is used to depict the car-following behavior of HDVs:

$$U_{\mathrm{IDM}} = U_{\max}\left[1 - \left(\frac{v_{\mathrm{FV}}}{v_e}\right)^4 - \left(\frac{g^*}{g}\right)^2\right] \tag{3.17}$$

where $a_{\max}$ is the maximum acceleration of AV, v_e is the expected velocity, and g is the gap between AV and HDV. g^* is the desired gap between AV and front HDV formulated as

$$g^* = d_s + v_{\mathrm{AV}}T_e - \frac{v_{\mathrm{AV}}\Delta v}{2\sqrt{u_{\max}b}} \tag{3.18}$$

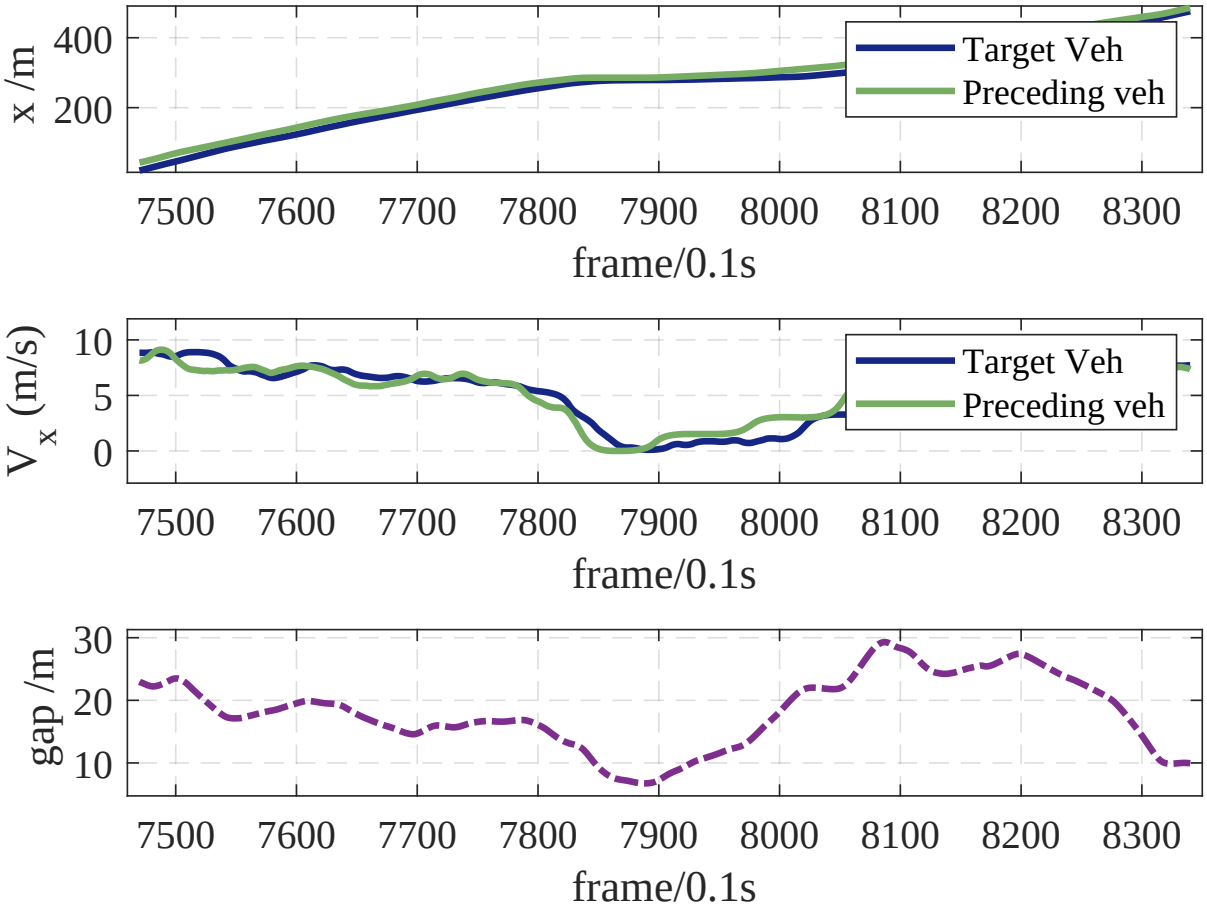


Figure 3.4: An Example of the IDM using the NGSIM.

where T_e is the expected time gap, Δv is the velocity difference between AV and FV, and b is the comfortable deceleration.

Fig. 3.4 illustrates a segment of the tracking process using the IDM with the Next Generation Simulation (NGSIM) dataset [200]. The green curve represents the target vehicle, while the blue curve indicates the preceding vehicle. The target vehicle uses the IDM to track the preceding vehicle. Initially, the target vehicle maintains a gap of 22.45 m and travels at a speed similar to that of the preceding vehicle. The IDM model accurately tracks the longitudinal position x and longitudinal velocity v_x of the preceding vehicle, showing only minor deviations. The gap between the target and preceding vehicles remains close to the desired distance of 15 m, demonstrating that the IDM model effectively maintains a safe and comfortable following distance.

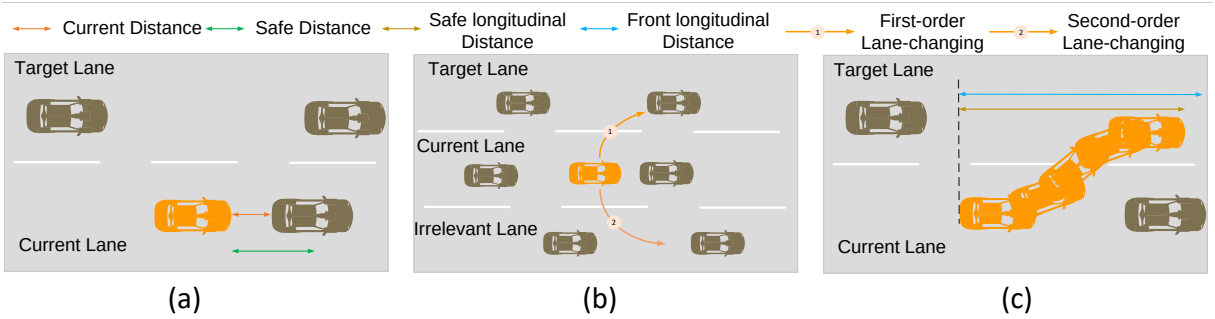


Figure 3.5: Selecting the proper driving lane under three scenarios.

3.5.2 The Lane-changing Rules of AV

Based on the relationships between the distance of the AV and the HDV and safety distance, this chapter separates lane-changing behaviors of the AV into two categories: passive lane-changing and active lane-changing. It is regarded as passive lane-changing when the AV performs the lane-change under the condition that $d_c < d_s$. Active lane-changing is considered under the condition that $d_f \geq d_s$, where d_f represents the distance between the AV and the HDV ahead in the target lane. In addition, the HDVs will perform passive lane changes with a defined probability of 10% to simulate and reflect the complex maneuvers encountered in real driving.

3.5.2.1 Passive Lane-changing

Fig. 3.5(a) presents a case where the AV performs passive lane-changing on dual-lane highways. In such a case, the AV would change lane when the following condition is satisfied:

$$f_{\mathrm{LC}} = \mathrm{True} \quad \text{if} \quad \exists d_c \leq d_s \tag{3.19}$$

where f_{LC} is the lane-changing judgment flag, which triggers the AV to changes the lane.

Fig. 3.5(b) presents a case where the AV performs passive lane-changing on three-lane highways when this AV drives in the middle lane. In this case, the first assessment is made to determine whether a safe lane-changing to the target lane can be made by the AV. If so, the AV will change to the target lane. Otherwise, it will change to the other lane.

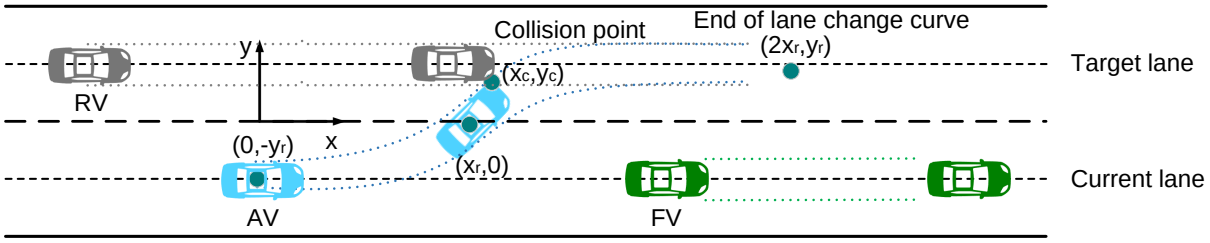


Figure 3.6: Collision detection in lane-changing.

3.5.2.2 Active Lane-changing

Fig. 3.5(c) presents a case where the AV performs active lane-changing on dual-lane highways. In such a case, the AV would perform lane-changing once the following condition is satisfied:

$$f_{\text{LC}} = \text{True} \quad \text{if} \quad \exists d_f \geq d_s \tag{3.20}$$

Similar to the passive lane-changing, on the three-lane highway, the AV first assesses whether the condition in (3.20) is satisfied in the target lane. If yes, the AV will change to the target lane. If not, the AV then assesses whether the condition in (3.20) is satisfied in the other lane. If yes, the AV will switch to the other lane. If neither the target lane nor the other lane meets the condition in (3.20), the AV will remain in its current lane.

this chapter primarily focuses on decision-making algorithms for autonomous driving. While real-world sensing errors, such as noise and delays, can impact lane-changing decisions, these challenges are typically addressed in the computer vision domain. The decision-making algorithms presented in this work were verified using a decision-making-focused simulator, consistent with prior research, where sensing uncertainties were not the primary focus. Future work could explore the integration of decision-making algorithms with camera systems and realistic driving equipment to better account for sensing errors in real-world scenarios.

3.5.3 Safety Evaluator

3.5.3.1 Detection of Collisions with RV

Fig. 3.6 presents the collision detection of lane-changing process, where the RV is assumed to travel at a constant velocity. This assumption could be attributed to the fact that drivers maintaining a constant speed are at the lowest risk of collisions [201]. The assumption that the RV maintains a constant speed during the collision detection process is applied specifically for decision-making within the training environment. This does not imply that the RV would maintain a constant speed in simulator. Instead, this assumption establishes a safety margin reference based on the current RV speed, which the neural network learns and adapts during training. The safety margin evolves flexibly, adjusting according to the relative speed and distance between the AV and RV. Moreover, this collision detection mechanism filters out unsafe scenarios, avoiding unnecessary training and improving training efficiency. This approach differs from traditional model-based methods, such as MPC, which rely on constant-speed assumptions for decision-making. In addition, a ideal and smooth lane-changing curve that conforms to human driving habits is used to model the lane-changing trajectory [202]. This curve is based on a cubic polynomial that describes the relationship between the lateral displacement and the longitudinal displacement. The cubic polynomial possesses the advantages of simplicity and smoothness while it can be easily optimized under various constraints and objectives [203]. The following cubic polynomial curve is used to perform the feasibility check:

$$y(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 \tag{3.21}$$

where c_0, c_1, c_2 and c_3 are the constant coefficients for the cubic polynomial cruve.

In Fig. 3.6, there are three positions labeled: $(0, -y_r)$, $(x_r, 0)$, and $(2x_r, 2)$, matching the start point, potential collision point, and final point respectively. To simplify the relationship between the lateral displacement and longitudinal displacement, the road width is assumed as 4 m and three positions are labelled: $(0, -2)$, $(x_r, 0)$, and $(2x_r, 2)$. The position of the potential collision point for the feasibility check is denoted by (x_c, y_c). Submitting the three positions into equation (3.21) gives $c_0 = -2$, $c_0 + c_1 x_r + c_2 x_r^2 + c_3 x_r^3 = 0$, and

$c_0+2c_1x_r+4c_2x_r^2+8c_3x_r^3=2$. These are simplified as $c_1=2c_3x_r^2$, $c_2=-3c_3x_r^2$ and $c_3=\frac{1}{x_r^3}$. Therefore, the cubic polynomials is formulated as

$$y(x)=-2+\frac{2}{x_r}x-\frac{3}{x_r^2}x^2+\frac{1}{x_r^3}x^3 \tag{3.22}$$

Inspired by [204], the time to the difference between potential collision position (TPCP) is used to assess the collision risks between AV and RV. It is computed by

$$\mathrm{TPCP_{HV}}=\frac{d_r}{v_{\mathrm{AV}}},\ \mathrm{TPCP_{RV}}=\frac{x_r}{v_{\mathrm{RV}}} \tag{3.23}$$

where $\mathrm{TPCP_{AV}}$ and $\mathrm{TPCP_{RV}}$ are the TPCP of AV and RV, respectively. d_r represents the distance from the starting position to the potential collision position. It can be expressed by

$$\begin{aligned} d_{\mathrm{r}} &= \int_0^{x_{\mathrm{r}}} \mathrm{d}l = \int_0^{x_{\mathrm{r}}} \sqrt{1+(y')^2}\,\mathrm{d}x \\ &= \int_0^{x_{\mathrm{r}}} \sqrt{1+\left(\frac{2}{x_{\mathrm{r}}}+\frac{6x}{x_{\mathrm{r}}^2}-\frac{6x^2}{x_{\mathrm{r}}^3}\right)^2}\,\mathrm{d}x \end{aligned} \tag{3.24}$$

where l represents the whole lane-changing curve. If the AV performs the lane-change maneuver without any collision with the RV, the following condition should be satisfied

$$|\mathrm{TPCP_{HV}}-\mathrm{TPCP_{RV}}|>T_{\mathrm{safe}} \tag{3.25}$$

where T_{safe} is a time parameter that represents the safety margin.

3.5.3.2 Detection of Collisions with FV

The TTC is used to assess the collision risk between AV and FV [205]. The TTC can be calculated as follows:

$$\mathrm{TTC}=\frac{l_{\text{AV-FV}}}{v_{\mathrm{AV}}-v_{\mathrm{FV}}} \tag{3.26}$$

where $l_{\text{AV-FV}}$ is the longitudinal distance from AV to FV.

Algorithm 2: Collision-Supervised Mechanism

Input: v_{AV}, v_{FV}, v_{RV}, d_{FV}, d_{RV},
T_{safe}: Safety time threshold,
$\mathrm{TTC}_{\mathrm{safe}}$: Safe time-to-collision threshold.
Output: Decision: Lane-changing or lane-keeping

```
TPCP_AV ← compute using (23)
TPCP_RV ← compute using (23)
if |TPCP_AV − TPCP_RV| > T_safe then
    TTC_AV ← compute using (26)
    if TTC ≥ TTC_safe then
        Decision ← Make lane-changing
    else
        Decision ← Make lane-keeping
else
    Decision ← Make lane-keeping
return Decision
```

If the AV performs the lane-change maneuver without any collision with the FV, the following condition should be satisfied

$$\mathrm{TTC} \geq \mathrm{TTC}_{\mathrm{safe}} \tag{3.27}$$

where $\mathrm{TTC}_{\mathrm{safe}}$ is the predefined threshold.

Based on the above description, the safety evaluation for lane-changing is summarized in Algorithm 2. This algorithm is based on the relative velocities and distances between the AV and both FV and RV in lane-changing. Once both conditions (25) and (27) are satisfied, the AV changes the lane. Otherwise, the AV keeps the lane. The final decision is returned as the output of the algorithm.

3.6 MPC for Enhancing DRL Performance

This section introduces robust control for the AV, including the vehicle dynamic model and MPC. The vehicle dynamic model simulates vehicle movements, enhancing the fidelity of the simulation. The MPC is to generate safe control commands, which is transferred from the upper-level decisions made by the RBDQN-CS. In particular, the MPC ensures AVs precisely and stably follow the actions determined by the risk-attention mechanism which is a part of the RBDQN-CS. The following control input and collision avoidance constraints are applied to ensure safety and feasibility:

$$\begin{aligned} & v_{\min} \leq v_{\mathrm{AV}}(t) \leq v_{\max}, u_{\min} \leq u(t) \leq u_{\max} \\ & \delta_{\min} \leq \delta(t) \leq \delta_{\max}, \|p_{\mathrm{AV}}(t) - p_{\mathrm{SV}}(t)\| \geq d_{\mathrm{safe}} \end{aligned} \tag{3.28}$$

where $v_{\min}$ and $v_{\max}$ denote the minimum allowable speed and maximum allowable speed for the AV, respectively. $u_{\min}$ and $u_{\max}$ denote the minimum allowable acceleration and maximum allowable acceleration for the AV, respectively. $\delta_{\min}$ and $\delta_{\max}$ denote the minimum allowable steering angle and the maximum allowable steering angle for the AV, respectively. $u(t)$ and $\delta(t)$ denote the acceleration and the steering angle for the AV at timestep t, respectively. At timestep t, the MPC controller finds the optimal solutions $u^*(t)$ and $\delta^*(t)$ by solving the optimization problem below:

$$\begin{aligned} \min J_c := & \sum_{k=0}^{N_p-1} (v_{\mathrm{AV}}(k) - v^*_{\mathrm{AV}})^2 + \\ & \sum_{k=0}^{N_p-1} \sum_{i=1}^{N_v} (\|p^i_{\mathrm{SV}}(k) - p^i_{\mathrm{AV}}(k)\| - d_{safe})^2 + \lambda \sum_{k=0}^{N_c-1} u^2(k) \\ \mathbf{s.t.}\ & 0 < v_{\mathrm{AV}}(k) \leq v_{\max}, u_{\min} \leq u(k) \leq u_{\max}, \\ & \delta_{\min} \leq \delta(k) \leq \delta_{\max}, SV \in N_V, k \in [0, N-1], \\ & v_{\mathrm{AV}}(k) \in \quad D_c \end{aligned} \tag{3.29}$$

Algorithm 3: MPC Controller for Vehicle Speed Adjustment

Input: v^*_{AV}: Desired speed for the autonomous vehicle
Output: $u[0]$: Optimal acceleration for the initial timestep, or fallback to the PID controller if no solution is obtained

```
/* Initialize MPC Controller and surrounding vehicle information     */
AV ← deepcopy(self) ;
N_V ← get_surrounding_vehicles() ;
opti ← ca.Opti() ;
u ← opti.variable(N) ;
J_c ← 0 ;
/* Formulate the cost function and constraints                       */
for k ← 0 to N − 1 do
    for vehicle in N_V do
        action ← use_RBDQN-CS_to_predict_vehicle_action(vehicle) ;
    δ(k) ← compute_steering(AV, N_V) ;
    AV.update(u(k), δ(k)) ;
    J_c ← J_c + (v_AV(k) − v*_AV)^2 ;
    for i ← 1 to N_sv do
        J_c ← J_c + (‖p^i_SV(k) − p^i_AV(k)‖ − d_safe)^2 ;
    J_c ← J_c + λ u^2(k) ;
    add_vehicle_constraints(AV, N_V, u(k)) ;
/* Solve the optimization problem                                    */
opti.minimize(J_c) ;
solution ← opti.solve() ;
```

where N_p and N_c denote the prediction horizon and control horizon, respectively. $v^*_{\mathrm{AV}}(k)$ is the target speed, N_sv is the number of surrounding vehicles, and λ is a weighting factor for the acceleration penalty. (30) represents AV's state at timestep t based on the following constraints of dynamics:

$$
\begin{aligned}
p_{\mathrm{AV}}(t+1) &= p_{\mathrm{AV}}(t) + v_{\mathrm{AV}}(t) \cdot \cos(h_{\mathrm{AV}}(t)) \cdot \Delta t \\
v_{\mathrm{AV}}(t+1) &= v_{\mathrm{AV}}(t) + u(t) \cdot \Delta t \\
h_{\mathrm{AV}}(t+1) &= h_{\mathrm{AV}}(t) + \frac{v_{\mathrm{AV}}(t)}{L} \cdot \tan(\delta(t)) \cdot \Delta t
\end{aligned}
\tag{3.30}
$$

where Δt is the timestep duration. L is the wheelbase length. $u(t)$ is the acceleration of the AV at timestep t. and $\delta(t)$ is the steering angle of the AV at timestep t.The entire control process is summarized in Algorithm 3.

3.7 Simulation Results

Table 3.1
Parameters for the RBDQN-CS Agent

Parameters	Default Settings	Description
n_episodes	5000	Total training episodes
gamma	0.99	Discount factor
train_steps	5	Train steps
learning_rate	0.001	Learning rate
batch_size	100	Batch size
eval_freq	200	Evaluation frequency
eval_episodes	5	Evaluation episodes
sim_frequency	15	Simulation frequency
policy_frequency	5	Policy frequency
low_level_control	True	Choose MPC
prediction_n_steps	7	Number of prediction steps

The proposed RBDQN-CS is validated by evaluating its convergence, converged value, as well as the collision rate and speed variation of the AV on the three-lane highways. To demonstrate the generalizability of the proposed algorithm, two different traffic flows are used. These two traffic flows involve 4-6 HDVs and 7-10 HDVs, referred to as the normal traffic flow and the high traffic flow, respectively. Among the whole validation, two kinds of verification are considered. In the first verification, the proposed RBDQN-CS is compared with four benchmarks: DQN, DQN with a balanced reward function (BDQN), DQN with a risk-attention mechanism (RDQN), and DQN with a collision-supervised mechanism (DQN-CS). In the second verification, the proposed RBDQN-CS is compared with four benchmark DRL algorithms under two different traffic flows. The four benchmark DRL algorithms include DDPG [38], PPO [34], Advantage Actor-Critic (A2C) [191] and DQN [190]. this chapter uses a highway virtual simulation platform called Highway-env to conduct the validation [206]. To illustrate the generalizability of the RBDQN-CS, the initial position and speed of HDVs are randomly set within a small range. The setting is achieved using Python 3.6, PyTorch 1.10.0, Ubuntu 20.04.6 LTS, a 12th generation 16-thread Intel® Core™ i5-12600KF CPU, an NVIDIA GeForce RTX 3090 GPU, and 64 GB of RAM. The environment setup and parameters used are detailed in Table 3.1. The `prediction_n_steps` parameter represents the number of future steps considered during trajectory prediction, commonly used in MPC frameworks. In this work,

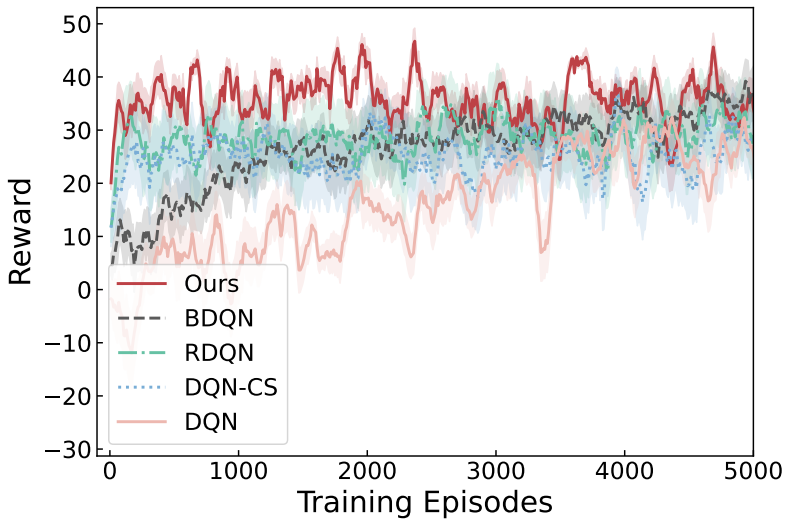


Figure 3.7: Rewards of the RBDQN-CS, BDQN, RDQN, DQN-CS, and DQN during the converging.

it guides decision-making by anticipating potential outcomes within a limited horizon. The choice of `7 steps` was determined empirically, balancing computational efficiency and prediction accuracy. During training, this prediction horizon ensured robust performance without causing any abrupt negative behavior. The robustness of decision-making relies not only on the prediction horizon but also on how effectively the agent utilizes these predictions to adapt its actions. Further fine-tuning of this parameter can be explored for more complex driving environments.

3.7.1 Comparison with DQN, RDQN, BDQN, and DQN-CS

Figs. 3.7 and 3.9 present the convergence, driving speed variation, collision rates of the proposed RBDQN-CS and RDQN, BDQN, DQN-CS, and DQN during the converging, respectively. Fig. 3.10 presents the rewards of the proposed RBDQN-CS and four other algorithms using converged policy. The other four algorithms include DQN, BDQN, RDQN, and DQN-CS. As shown in Fig. 3.7, the proposed RBDQN-CS reaches a higher reward and faster convergence compared to four other algorithms. The rewards of the proposed RBDQN-CS, BDQN, RDQN, DQN-CS and DQN, are around 35, 30, 27, 25, and 20, respectively. As shown in Fig. 3.8, the proposed RBDQN-CS achieves a higher average speed and faster convergence compared to four other algorithms. The average speed and variance of the RBDQN-CS, BDQN, RDQN, DQN, and DQN-CS are (22, 2), (20, 3), (20,

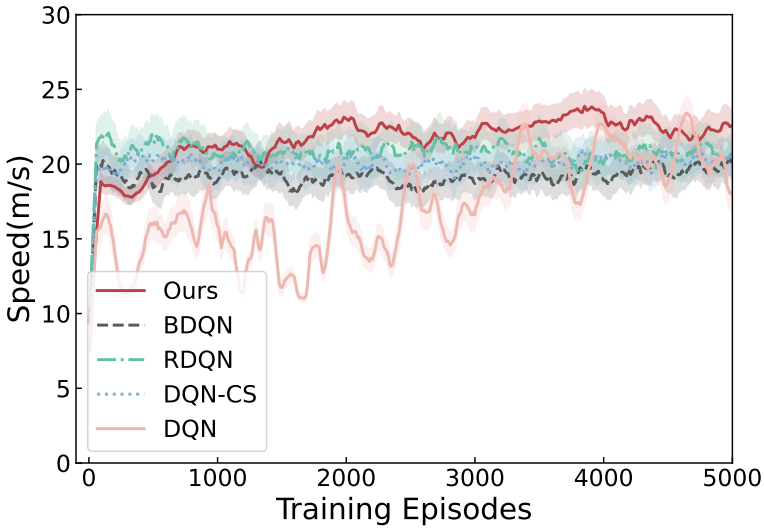


Figure 3.8: Speed variations of the RBDQN-CS, BDQN, RDQN, DQN-CS, and DQN during the converging.

3), (18, 5), and (18, 5), respectively. As shown in Fig. 3.9, the proposed RBDQN-CS has a lower collision rate and faster convergence. The collision rate of the proposed RBDQN-CS, BDQN, RDQN, DQN, and DQN-CS are below 0.05 per 100 episodes, 0.10 per 100 episodes, 0.10 per 100 episodes, 0.15 per 100 episodes, and 0.15 per 100 episodes, respectively. As shown in Fig. 3.10, the proposed RBDQN-CS has a higher reward after the converging compared to four other algorithms. The rewards of the proposed RBDQN-CS, BDQN, RDQN, DQN-CS and DQN, are around 40, 30, 30, 27, and 24, respectively. The above analysis demonstrates the effectiveness in terms of the safety and efficiency of the proposed RBDQN-CS.

Higher rewards indicate better balance across key evaluation factors, including safety, efficiency, and unselfishness. In dynamic scenarios, higher rewards reflect successful decision-making, such as safe lane changes and efficient driving. Lower rewards can suggest conservative behavior, prioritizing safety in complex situations. The priority-based reward tuning used in this work ensures that higher rewards align with real-world driving needs, emphasizing safety first, followed by efficiency and unselfishness. This approach prevents aggressive driving while promoting adaptive and context-aware decision-making.

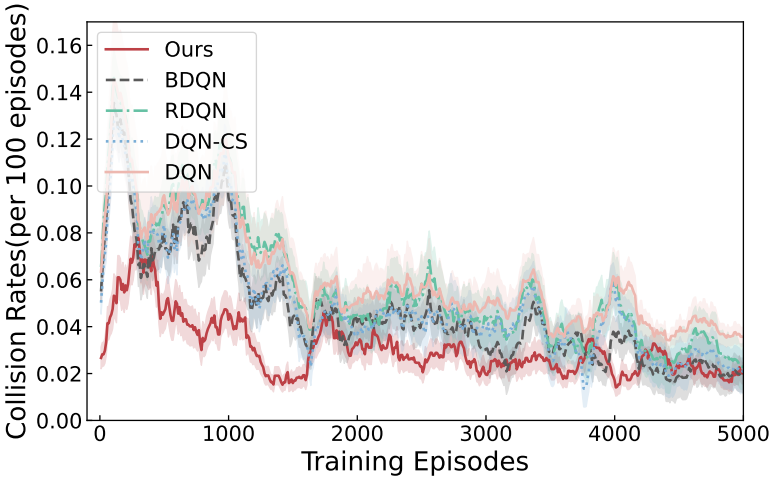


Figure 3.9: Collision rates of the RBDQN-CS, BDQN, RDQN, DQN-CS, and DQN during the converging.

3.7.2 Comparison under Different Traffic Flows

Figs. 3.11 and 3.13 present three indicators of the proposed RBDQN-CS and the four benchmark DRL algorithms over 5000 training episodes in the normal and high traffic flows, respectively. The three indicators include convergence, speed variation, and collision rates. The four benchmark DRL algorithms include DQN, PPO, A2C, and DDPG. Figs. 3.12 and 3.14 present the rewards of the proposed RBDQN-CS and the four benchmark DRL algorithms in the normal and high traffic flows using converged policy, respectively.

3.7.2.1 Normal Traffic Flow

As shown in Fig. 3.11(a), the proposed RBDQN-CS reaches a higher reward and faster convergence compared to the four benchmark DRL algorithms. The rewards of the proposed RBDQN-CS, DQN, A2C, PPO and DDPG, are around 37, 23, 20, 18, and 17, respectively. As shown in Fig. 3.11(b), the proposed RBDQN-CS achieves a higher average speed and faster convergence compared to the four benchmark DRL algorithms. The average speed and variance of the RBDQN-CS, DQN, DDPG, PPO, and A2C are around (24, 2), (22.5, 1.5), (22, 5), (20, 3), and (19, 3), respectively. As shown in Fig. 3.11(c), the proposed RBDQN-CS has a lower collision rate and faster convergence. The collision rates of the proposed RBDQN-CS, DQN, A2C, DDPG, and PPO are below 0.02 per 100 episodes, 0.03 per 100 episodes, 0.035 per 100 episodes, 0.04 per 100 episodes, and

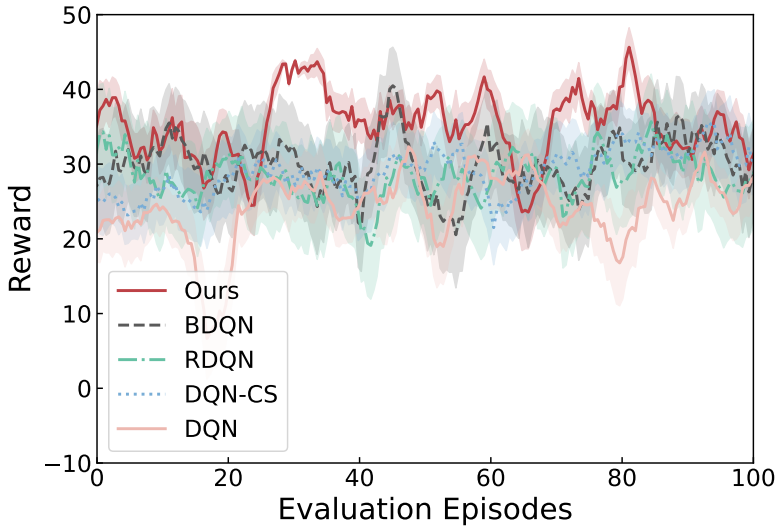


Figure 3.10: Reward of the RBDQN-CS, BDQN, RDQN, DQN-CS, and DQN after the converging.

0.045 per 100 episodes, respectively. As shown in Fig. 3.12, the proposed RBDQN-CS has a higher reward after the converging compared to the four benchmark DRL algorithms. The rewards of the proposed RBDQN-CS, DQN, A2C, PPO and DDPG, are around 38, 23, 20, 19, and 18, respectively.

3.7.2.2 High Traffic Flow

As shown in Fig. 3.13(a), the proposed RBDQN-CS reaches a higher reward and faster convergence compared tothe four benchmark DRL algorithms. The rewards of the proposed RBDQN-CS, PPO, DQN, A2C, and DDPG, are around 35, 25, 20, 10, and 5, respectively. As shown in Fig. 13(b), the proposed RBDQN-CS achieves a higher average speed and faster convergence compared to the four benchmark DRL algorithms. The average speed and variance of the The average speed and variance of the RBDQN-CS, DQN, A2C, DDPG, and PPO are around (22.5, 3.5), (18, 6), (17.5, 3), (17, 1.5), and (16, 3), respectively. As shown in Fig. 13(c), the proposed RBDQN-CS has a lower collision rate and faster convergence. The collision rates of the proposed RBDQN-CS, DQN, A2C, DDPG, and PPO are below 0.04 per 100 episodes, 0.06 per 100 episodes, 0.08 per 100

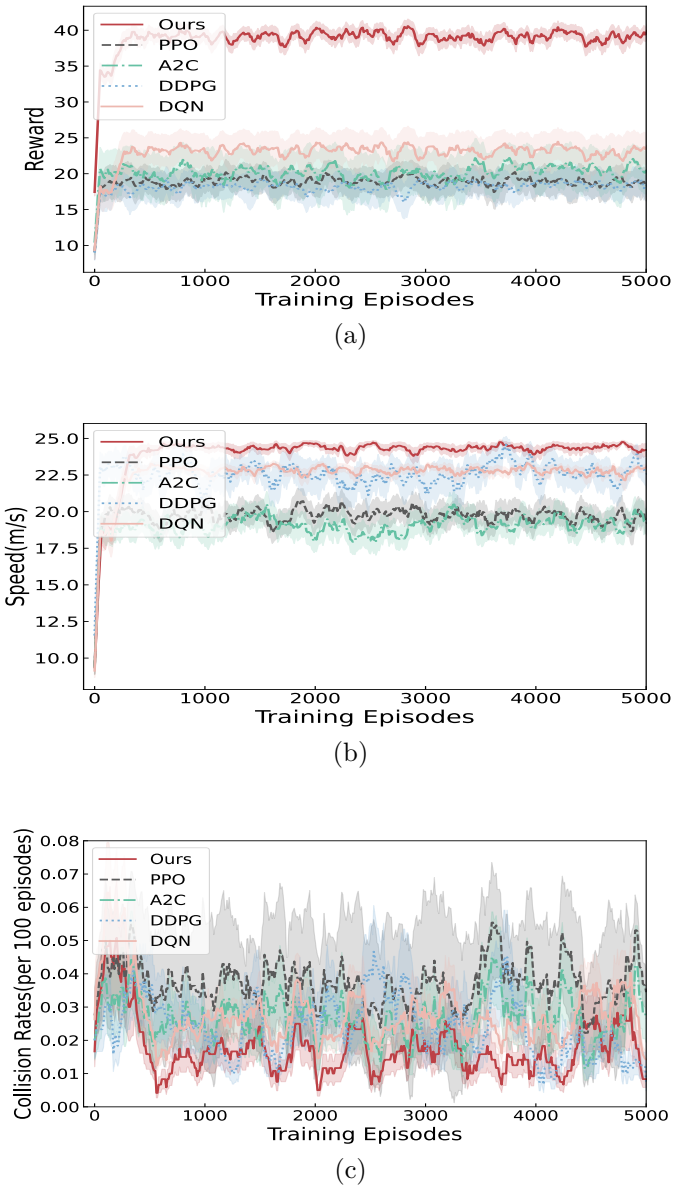


Figure 3.11: Performance of the RBDQN-CS, PPO, A2C, DDPG, and DQN during convergence in normal traffic flow. (a) Rewards; (b) Speed variations; (c) Collision rates.

episodes, 0.11 per 100 episodes, and 0.12 per 100 episodes, respectively. As shown in Fig. 3.14, the proposed RBDQN-CS has a higher reward after the converging compared to the four benchmark DRL algorithms. The rewards of the proposed RBDQN-CS, DDPG, DQN, A2C and PPO are around 40, 27, 24, 8, and 3, respectively.

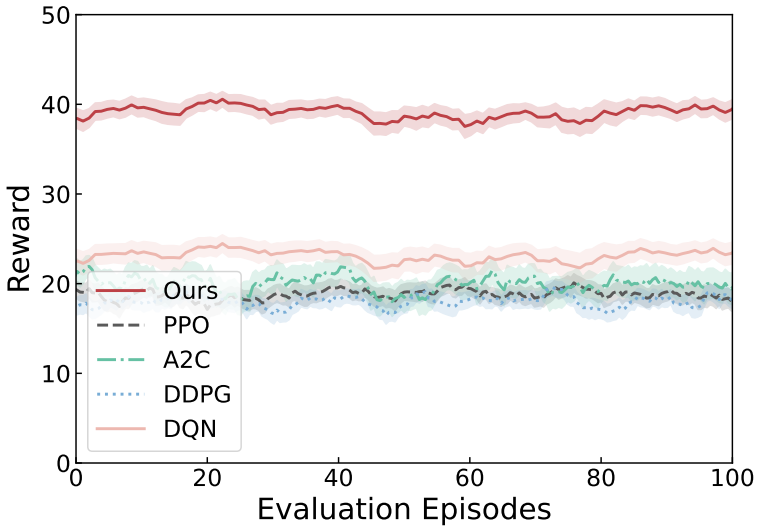


Figure 3.12: Rewards of the RBDQN-CS, PPO, A2C, DDPG, and DQN in the normal traffic flow after the converging.

3.7.2.3 Speed and Collision Rate after the Converging

after the converging, the comparison of speed variation and collision rate between the proposed RBDQN-CS and the four benchmark DRL algorithms are made in both normal and high traffic flows. The results are presented in Table 3.2. In the normal traffic flow, the RBDQN-CS demonstrates the lowest collision rate at 2.3%, followed by DDPG at 4.1%, A2C at 4.3%, and 4.6%, indicating the safety performance of the RBDQN-CS. In addition, the RBDQN-CS has the highest average speed of 24.73 m/s, which is larger than that of PPO at 20.36 m/s, A2C at 18.83 m/s, DDPG at 21.86 m/s, and DQN at 22.34 m/s, showcasing greater efficiency. In the normal traffic flow, the collision rate of the RBDQN-CS is at 3.4%, which is smaller than that of PPO at 11.6%, A2C at 7.3%, DDPG at 8.2%, and DQN at 8.3%. In addition, the average speed of the RBDQN-CS is 22.36 m/s,which is greater than PPO at 16.67 m/s, A2C at 18.04 m/s, DDPG at 17.73 m/s, and DQN at 19.33 m/s.

The marginal differences observed in the highlighted average speed values can be attributed to the controlled simulation environment, where stable traffic conditions limit performance variation once collision avoidance and efficient lane-changing are achieved. To further improve speed performance, strategies such as traffic-aware reward shaping, which adjusts rewards based on traffic density, and adaptive speed planning, where the

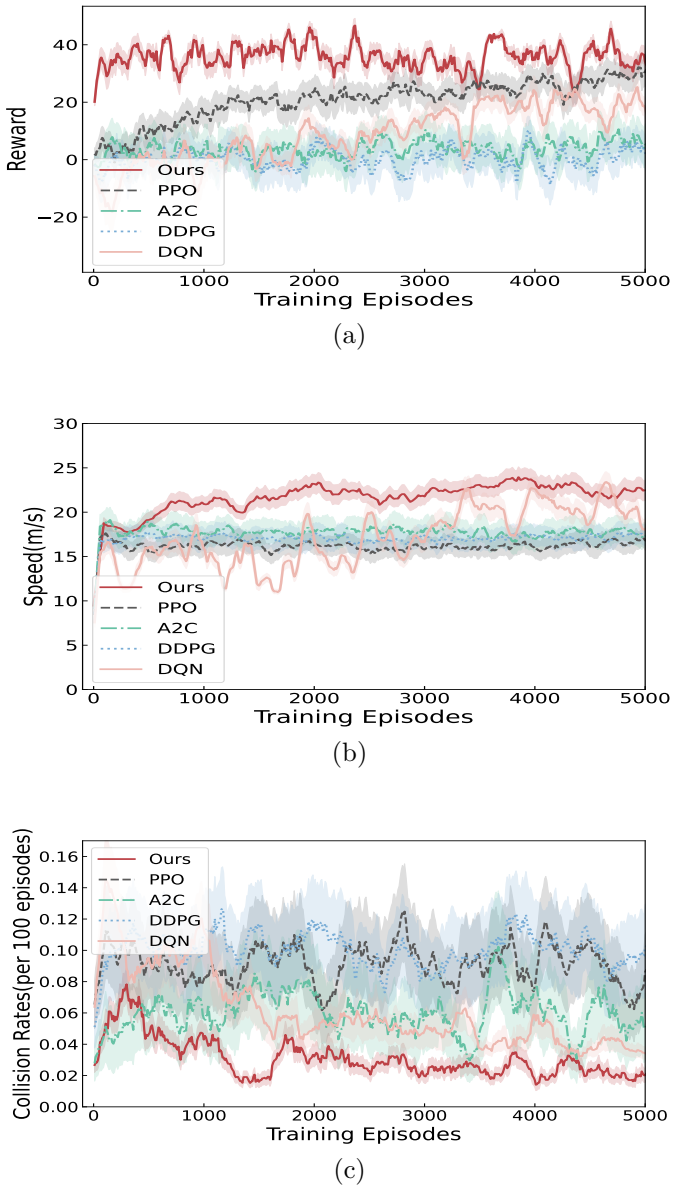


Figure 3.13: Performances of the RBDQN-CS, PPO, A2C, DDPG, and DQN during convergence in high traffic flow. (a) Rewards; (b) Speed variations; (c) Collision rates.

agent optimizes speed according to surrounding vehicle behavior, can be applied. Additionally, enhanced exploration techniques, like curiosity-driven learning, could help the agent discover more efficient driving patterns. While the current results reflect a balanced trade-off between speed and safety, these refinements could further enhance the advantages of the RBDQN-CS algorithm under more dynamic driving conditions. In summary, when converging, the proposed RBDQN-CS has higher rewards, higher average speed, a lower collision rate and a higher reward using converged policy than the four benchmark DRL algorithms in the normal and high traffic flows. After the converging, the proposed

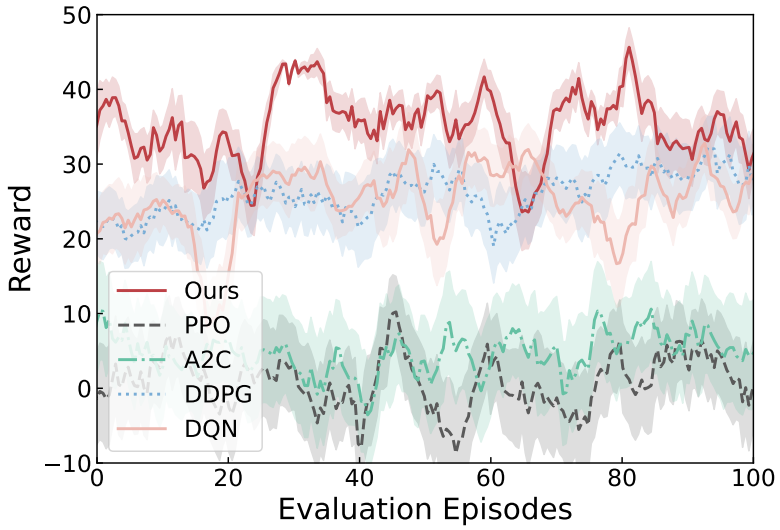


Figure 3.14: Rewards of the RBDQN-CS, PPO, A2C, DDPG and DQN in the high traffic flow after the converging .

Table 3.2
Comparison of Collision Rate and Average Speed after the Converging

Scenarios	Metrics	PPO	A2C	DDPG	DQN	Ours
Normal traffic Flow	coll. rate (%)	4.6	4.3	4.1	3.7	**2.3**
	avg. v (m/s)	20.36	18.83	21.86	22.34	**23.73**
High traffic Flow	coll. rate (%)	11.6	7.3	8.6	8.2	**3.4**
	avg. v (m/s)	16.67	18.04	17.73	19.33	**22.36**

coll. rate means collision rate per 100 episodes during the converging. The best results are highlighted in bold.

RBDQN-CS has higher average speed and lower collision rate than the four benchmark DRL algorithms in the normal and high traffic flows. The results demonstrate the effectiveness in terms of the safety and efficiency of the proposed RBDQN-CS in the normal and high traffic flows. While the RBDQN-CS algorithm shows improved performance in both average speed and collision reduction compared to benchmark DRL algorithms, achieving near-perfect safety and avoiding almost all collisions requires further enhancements. This can be addressed by refining the existing DQN-based framework with advanced strategies tailored for highway driving. These include risk-aware planning with safety shields, adaptive safety margins that adjust based on traffic density, and multi-modal sensor fusion to improve environmental perception. Additionally, adaptive reward shaping can further pe-

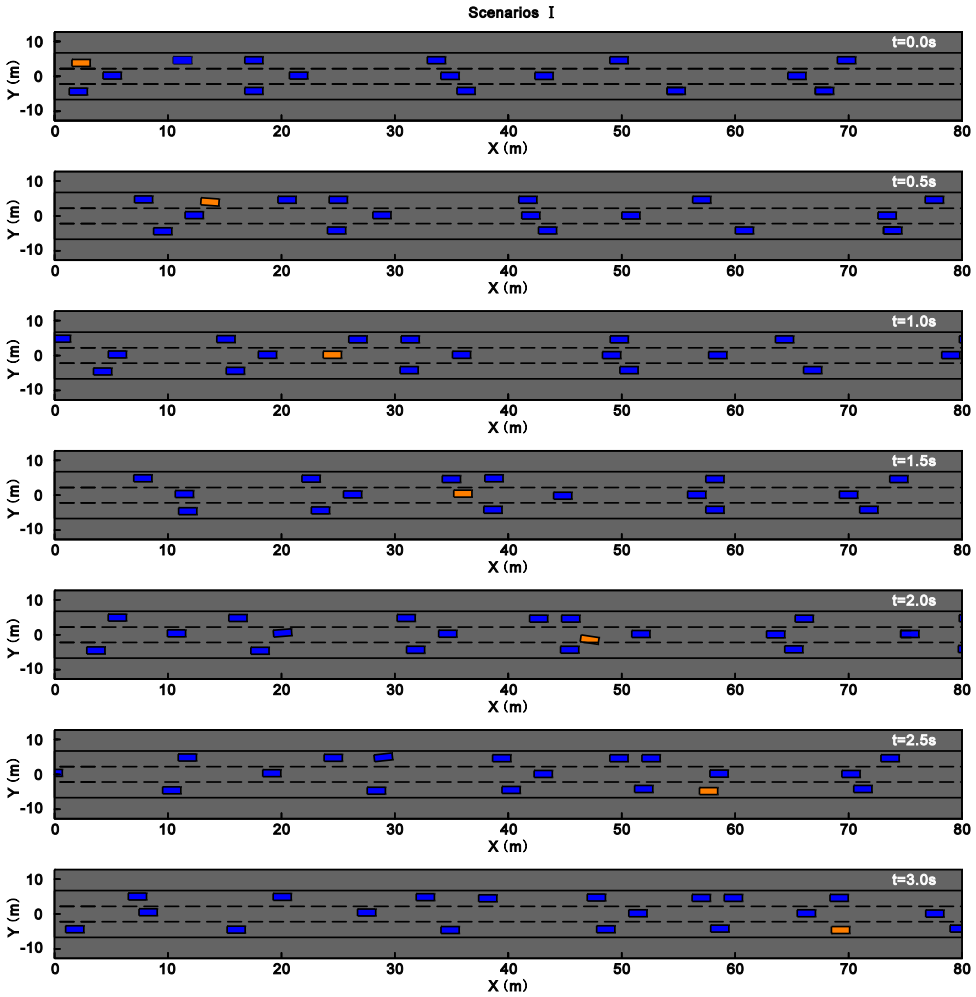


Figure 3.15: Example illustration in the normal traffic flow.

nalize near-collision events, while continuous learning through online updates or transfer learning can enhance adaptability to evolving traffic patterns. These improvements would strengthen the robustness of the RBDQN-CS framework while maintaining efficient and safe highway driving.

3.7.3 Examples in the Normal and High Traffic Flows

Using the proposed RBDQN-CS, the AV successfully changes the lane. Figs. 3.15 and 3.16 present lane-changing of the AV when it interacts with HDVs between $0-3$ s in the normal and high traffic flows, respectively. The orange and blue blocks represent the AV and the HDV, respectively. During this process, the AV changes lanes in three-lane highways twice and seven time points of the AV are presented, including $t = 0$ s, $t = 0.5$ s, $t = 1.0$ s, $t = 1.5$ s, $t = 2.0$ s, $t = 2.5$ s, and $t = 3.0$ s. The three lanes of this highway are named by the upper, middle and bottom lanes from top to bottom.

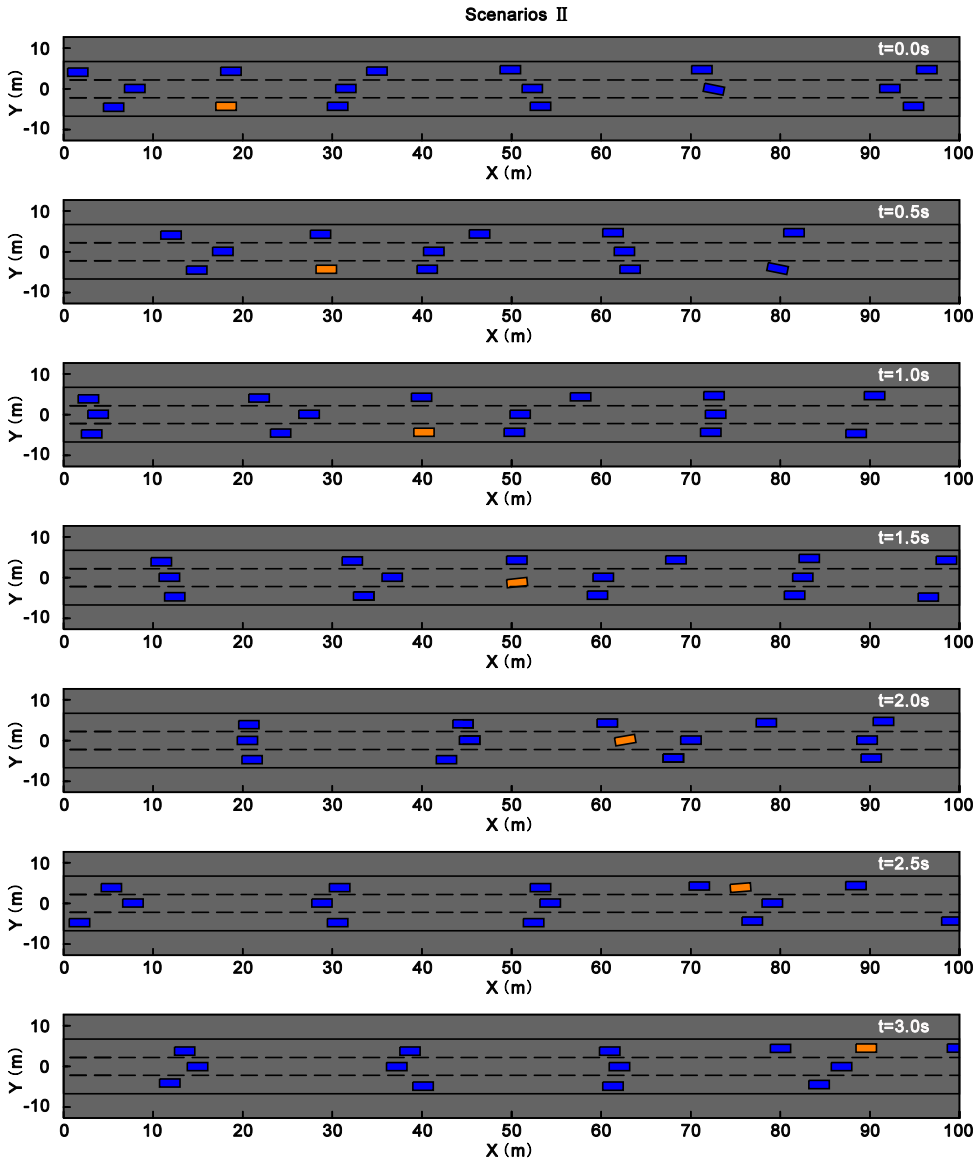


Figure 3.16: Example illustration in the high traffic flow.

In the normal traffic flow, when $t = 0.5$ s and $t = 2.0$ s, the AV starts changing the lane for the first time (from the upper to middle lane) and for the second time (from the middle to bottom lane), respectively. At $t = 1.0$ s and $t = 2.5$ s, the AV changes lanes for the first and second time, respectively.

In the high traffic flow, when $t = 1.0$ s and $t = 2.0$ s, the AV starts changing the lane for the first time (from the bottom to middle lane) and for the second time (from the middle to upper lane), respectively. At $t = 1.5$ s and $t = 2.5$ s, the AV changes lanes for the first and second time, respectively. In both traffic flows, the AV maintains stable and safe aside from the lane-changing.

3.8 Evaluation Based on DDTUI

The RBDQN-CS has considered and verified all the factors of DDTUI, contributing to state-of-the-art DRL-based decision-making. For driving safety, RBDQN-CS employs a collision penalty to ensure that the safety intentions generated during convergence and collision detection function as rule-based safety mechanisms, effectively reducing collisions. Verification is achieved by comparing the collision rates of RBDQN-CS with those of other benchmark algorithms, indicating lower collision rates. The rule-based safety mechanism provides comprehensible mathematical formulations, enhancing interpretability. Regarding driving efficiency, RBDQN-CS uses an efficiency reward to encourage the AV to reach the endpoint as quickly as possible. Verification is achieved by comparing the average speeds of RBDQN-CS with other benchmark algorithms, demonstrating higher average speeds. For training efficiency, RBDQN-CS utilizes a risk-attention mechanism, a balanced reward function, and a collision supervisor to achieve faster convergence. Verification is achieved by comparing the convergence rates of RBDQN-CS with other benchmark algorithms, indicating faster convergence. Unselfishness is addressed by limiting the number of lane changes to minimize disturbance to other vehicles. Verification is provided through scenarios illustrating the AV's minimal lane changes from the initial lane to the target lane. RBDQN-CS considers the needs of users, companies, and public traffic, supporting the adaptation of DRL-based decision-making as a practical real-world solution.

3.9 Summary

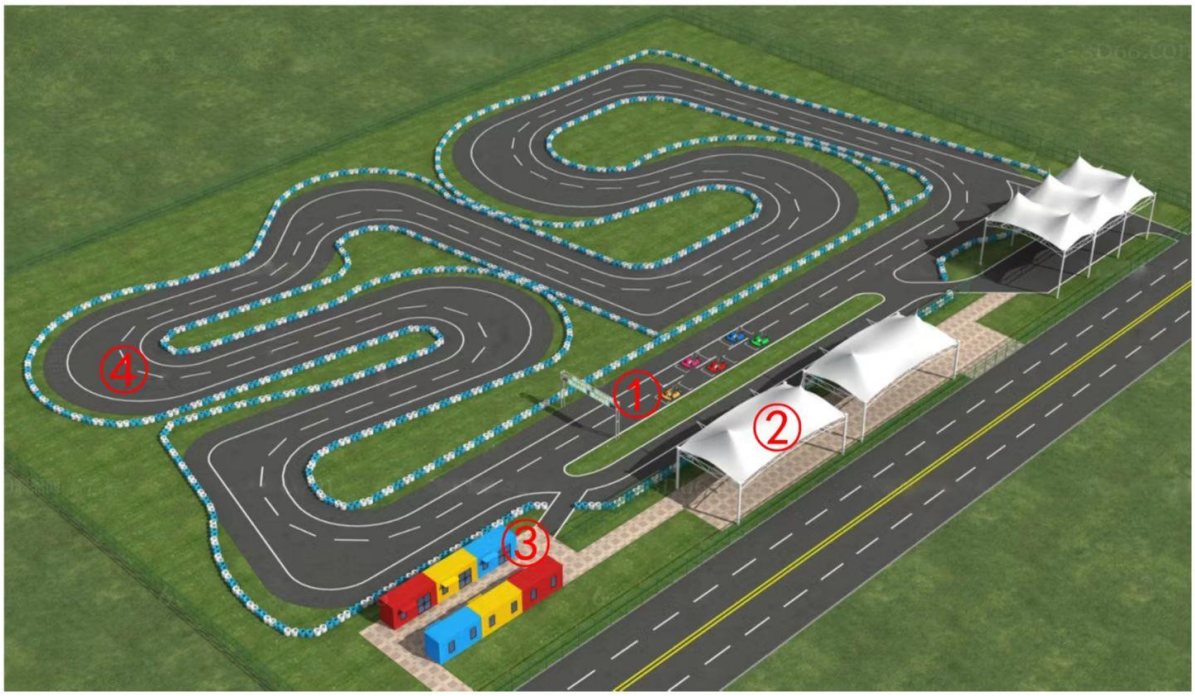
this chapter proposed a DRL algorithm, called RBDQN-CS, to improve the safety and efficiency of AVs on highway interactive driving. This proposed RBDQN-CS is achieved by introducing the risk-attention mechanism, the balanced reward function, and the collision-supervised mechanism to DRL. The effectiveness of this proposed RBDQN-CS with respect to safety and efficiency has been validated. In particular, the proposed RBDQN-CS surpasses DQN, BDQN, RDQN, and DQN-CS in terms of a higher reward both at convergence and after convergence, lower collision rate, higher average speed, and faster convergence. In addition, The proposed algorithm outperforms PPO, A2C, DDPG, and

DQN in terms of a higher reward both at convergence and after convergence, lower collision rate, higher average speed, and faster convergence in the normal and high traffic flows. In the future, extensive research will be conducted in two aspects, including 1) improving the predictions for HDV's trajectories, and 2) extending the algorithm to tasks with multi AVs by using multiple agents.

Chapter 4

Balanced Exploration and Curiosity-inspired Decision Making

Autonomous racing is a specialized case of autonomous driving. Compared to standard autonomous driving, autonomous racing requires decision-making at the extreme situations, increasing its difficulty. However, full exploration of extreme-case driving benefits the development of autonomous driving, as extreme cases reveal the upper capabilities of autonomous systems. While Chapter 4 proposed a decision-making framework for standard autonomous driving, this chapter explores decision-making under these extreme conditions. Autonomous racing has attracted extensive interest due to its great potential in self-driving at the extreme limits. Model-based and learning-based methods are widely used in autonomous racing. Model-based methods often struggle in complex environments when only local perception is available. This limitation can be overcome by Proximal Policy Optimization (PPO), a typical learning-based method that does not rely heavily on global perception. PPO is an on-policy reinforcement learning algorithm that improves training stability using a clipped surrogate objective to limit the change in policy between updates, ensuring robust and efficient learning. However, existing PPO faces challenges with low training efficiency in long sequences. To solve this issue, this chapter develops an improved PPO by introducing a curiosity mechanism, a balanced reward function, and an image-efficient actor-critic network. The curiosity mechanism focuses on training on key segments, facilitating efficient short-term learning of the PPO. The balanced reward function adjusts rewards based on the complexity of racetracks, promoting efficient exploration of the control strategy during training. The image-efficient actor-critic network



① Start/End point ② Rest Area ③ Preparation Region ④ Racing Lane

Figure 4.1: Sketch of a closed-circuit car racing environment.

enhances the PPO to fast process the perceived information. Simulation results on a physical engine demonstrate that the proposed algorithm outperforms benchmark algorithms in achieving less number of collisions, higher peak reward with less training time, and shorter laptime among multiple testing racetracks.

4.1 Introduction

Racing is a challenging and exciting sport that requires reliable decision making, precise control, and robust perception because of complex racetracks. As illustrated in Fig. 4.1, the racetracks are designed with a series of sharp bends, which makes safe driving more difficult at high velocity.

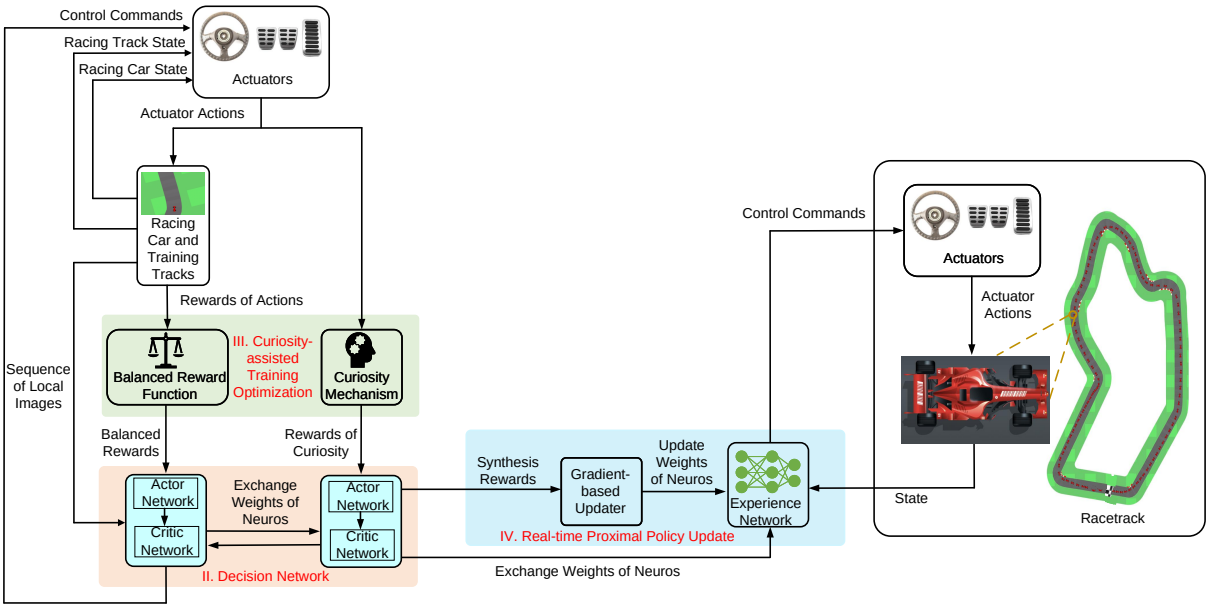


Figure 4.2: Diagram of the autonomous racing algorithm using the curiosity-assisted proximal policy optimization.

To address the limitations of traditional car racing approaches, autonomous racing has been developed, which combines the excitement of human car racing and the state-of-the-art autonomous driving technologies. Compared to traditional car racing, autonomous racing can drive through complex tracks at the speed limits with high precision due to its superior decision-making capabilities. The capabilities of autonomous racing have been demonstrated in the Roborace [207–209], Indy Autonomous Challenge [210–212], and Formula Student Driverless [213].

Global perception and local perception are both being applied to the autonomous racing. Out of which, local perception-based methods rely less on equipment and therefore are more cost effective. Perception-based decision making consists of model-based and learning-based methods. Learning-based methods are more promising because the global perception is not excessively used. For example, RL is capable to adapt to the local conditions of the environment and generates optimal control commands [25]. DRL, extending RL with DNN to handle complex functions, allows agents to learn from high-dimensional inputs like images. Existing DRL algorithms, such as the PPO, perform well in short-term gaming scenarios. However, these algorithms still encounter challenges in the learning during the long-duration racing. To this end, a local perception-based, image-efficient, and balanced reward-orientated PPO with curiosity mechanism (PPO-C) is proposed in this

chapter, as illustrated in Fig. 4.2. The inputs of the decision network are the sequence of local images. An image-efficient decision network is proposed to process images and generate safe control commands. The curiosity mechanism [214] uses intrinsic rewards to encourage the agents paying more attention to the steps with large prediction errors. Therefore, the agents can mitigate the uncertainties in local planning. Furthermore, a balanced reward function is proposed to consider both historical and prospective actions. The main contributions of this chapter are as follows.

- Only the local perception is used to get images that combine the racing vehicle and the surrounding environment. Global perception is no more required in detecting the boundaries and the center line of the racing track.
- The time required to reach the saturation value of rewards is significantly reduced, and the collisions with sharp bends are avoided. The convergence of the training is improved over benchmark algorithms.
- Shorter laptime and less collisions are achieved by the proposed balanced reward function. The challenge of maintaining balanced exploration over long sequences is tackled by introducing the balanced reward function.

4.2 Related Works

4.2.1 Challenges in Autonomous Racing

In traditional car racing, human driving skills dominate the competition because unexpected disturbances are often encountered. To minimize the effects of these disturbances, two main approaches have been developed. The first approach aims to optimize the aerodynamics of the racing car, and the second approach is to design effective control strategies. Despite the demonstrated effectiveness, the first approach is restrained by the limited potential for improvement. For the majority of race cars, the aerodynamic models have been optimized to their maximum capacity. The drawback of the second approach lies in the absence of an experience-based decision-making mechanism. Therefore, the control performance cannot be effectively transferred to different tracks. Existing methods in autonomous racing are mainly ground in global perception and local perception. Global

Table 4.1: Enhanced comparison of key features across different reinforcement learning algorithms with reasons for differences

Feature	**PPO-C**	**DDPG**	**PPO**	**SAC**	**Reasons for Difference**
Enhanced convergence consistency	✓		✓		PPO-C and PPO use a clipped objective function limiting excessive updates, enhancing training convergence consistency.
Ability to focus on complex segments	✓				Curiosity rewards enhance learning in complex scenarios against SAC and DDPG.
Adaptation to complex environments	✓				Curiosity rewards help PPO-C adjust strategy more effectively in complex conditions than other algorithms.
Improved data utilization efficiency	✓		✓		PPO-C and PPO update their learning multiple times per sample, improving efficiency.
Optimization of critical behaviors	✓		✓	✓	PPO-C targets critical areas through intrinsic curiosity, unlike standard PPO or SAC.
Promoting exploration in uncertainty	✓				Curiosity-based exploration targets high uncertainty areas ignored by other algorithms.
Advanced reward structure	✓				PPO-C uses prediction errors in rewards to accelerate learning, unlike other algorithms.

perception leverages comprehensive environmental data, the whole maps, and precise localization to provide a broad context for long-term planning [215, 216]. External sensors have been applied to global perception such as GPS, Inertial Measurement Unit (IMU), or Vehicle-to-Everything (V2X) communication. On the other hand, local perception focuses on real-time sensor data to detect and respond to immediate surroundings, ensuring dynamic object detection, short-term planning, and collision avoidance. Local perception uses onboard sensors, such as cameras and LIDARs [217]. For example, local perception is employed to perceive the surrounding road geometry and plan the vehicle speed in high-speed driving [218].

Global perception-based methods, predominately used in real world racing, heavily depend on specific perception conditions [212, 219, 220]. However, local perception-based methods are not bounded by specific perception conditions, reducing costs associated with global perception-based equipment. Therefore, local perception-based methods have gained popularity in autonomous racing [221, 222]. Model-based methods rely on pre-defined models or extra processes, such as Gaussian Process (GP) to quantify uncertainties [223]. However, model-based methods are incapable to cope with complex environments when only local perception is available. Model-based methods struggle in complex environments with only local perception due to their reliance on predefined planning and optimization rules. Without global information, these methods often lack the flexibility to handle unpredictable sections, as they may not obtain safe and efficient routes in unseen environments. A path-planning method is proposed in [224] that uses a path created by connecting the center lines on the straights and using clothoids between the center lines. The forward center line is required for global perception. A minimum-time optimal control problem using the centerline of the racetrack is formulated in [225]. Furthermore, uncertainty quantification in Model-based methods, such as GP, may encounter challenges when the real racetrack differs significantly from the tracks used to define the uncertainties. As a comparison, learning-based methods learn the optimal driving manner from data [226]. DRL, an advanced learning-based method that leverages deep neural networks to approximate complex functions, enables agents to learn from locally perceptive images. Additionally, [227] secured the world championship in automobile racing by using the DRL. It demonstrated the outstanding capability of DRL to enhance both the safety and stability in autonomous racing. Furthermore, [228] proposes a DRL powered racing system that surpasses the quickest human driver among a dataset comprising more than 50,000 players.

4.2.2 Deep Reinforcement Learning

State-of-the-art results of using DRL have been demonstrated in autonomous cars [208]. Recently, a set of DRL algorithms with exceptional performance have attracted interest, such as DDPG, SAC and PPO algorithms.

DDPG is an off-policy algorithm that uses deep neural networks to learn the control policy. With its suitability for handling high-dimensional data, multiple demonstrations of using DDPG have been presented in autonomous driving [38]. In particular, a DDPG model was proposed for safe driving within an end-to-end architecture [38]. Improved DDPG models have been proposed to enhance training efficiency [39]. The speed of racing cars could be accelerated by using DDPG, as demonstrated in [229]. A vision-based DDPG that considers driving safety at high speeds was proposed in [230]. In these studies, DDPG produces a definite control policy instead of a probability distribution of control policies. However, this definite control limits the exploration of other potential actions, implying that the decision may be satisfactory but not optimal. SAC is another off-policy model that incorporates a maximum entropy framework to enhance training robustness [231]. It has been shown to achieve higher average speeds than DDPG on multiple racetracks [232]. However, [232] focuses solely on optimizing average speed without considering other factors, such as reducing collisions with racetracks. Although SAC encourages exploration, it might not efficiently explore strategies to simultaneously minimize lap times and avoid collisions due to its undirected exploration. Furthermore, SAC's entropy term in the loss function sometimes leads to excessive exploration and slower convergence in complex scenarios.

The PPO depicts the control policy as a probability distribution, which facilitates faster exploration of strategies compared to DDPG [195]. Moreover, PPO uses a policy gradient-based method, achieving a stable equilibrium and providing assurance of its steadiness [233]. In contrast, off-policy algorithms are unstable and ineffective because they rely on training data that must be efficient under the current policy [234]. PPO has been used for generating driving strategies that balance safety and efficiency [34]. However, PPO aims to identify the most favorable steps for improvement while avoiding regression that could lead

to performance degradation. In PPO, the agent may struggle to generalize its experiences across different states and actions, leading to slower convergence. Moreover, PPO is prone to falling into local optima, which increases the training time [235]. Furthermore, the training efficiency of PPO in complex environments is low [236].

In summary, current DRL algorithms encounter challenges in fully exploring the environment, unstable training in off-line algorithm, and exhibiting lower convergence speed. PPO addresses some of these challenges by employing probability distributions for exploration. However, the training efficiency of PPO diminishes in complex-environment tasks like racing due to the increased variability. Moreover, the inherent risk of collisions with track boundaries during perilous turns remains unresolved due to its averaged intention mechanism. Additionally, achieving balanced exploration is crucial in long sequences to effectively construct the probability distribution of PPO. To mitigate these issues, a balanced reward-orientated PPO with curiosity mechanism is proposed in this chapter. The proposed curiosity mechanism directs the attention of PPO to critical short segments, thus enhancing the training efficiency. Furthermore, the balanced reward function facilitates balanced exploration from a global perspective. As a result, the low convergence speed and poor performance in crucial racing sections of PPO are addressed by introducing the curiosity mechanism and the balanced reward function.

The advantages of PPO-C compared to PPO, DDPG, and SAC are summarized in Table 4.1. PPO-C uses intrinsic rewards to drive targeted exploration towards less-understood regions. The targeted exploration is particularly beneficial in complex environments such as racing, where standard rewards are sparse or less informative. Moreover, PPO-C excels in dynamically changing environments by continually adapting its policy to maximize both normal and intrinsic rewards. The adaptive learning fosters learning in crucial and difficult-to-navigate parts, optimizing critical behaviors, and promoting exploration based on state uncertainty.

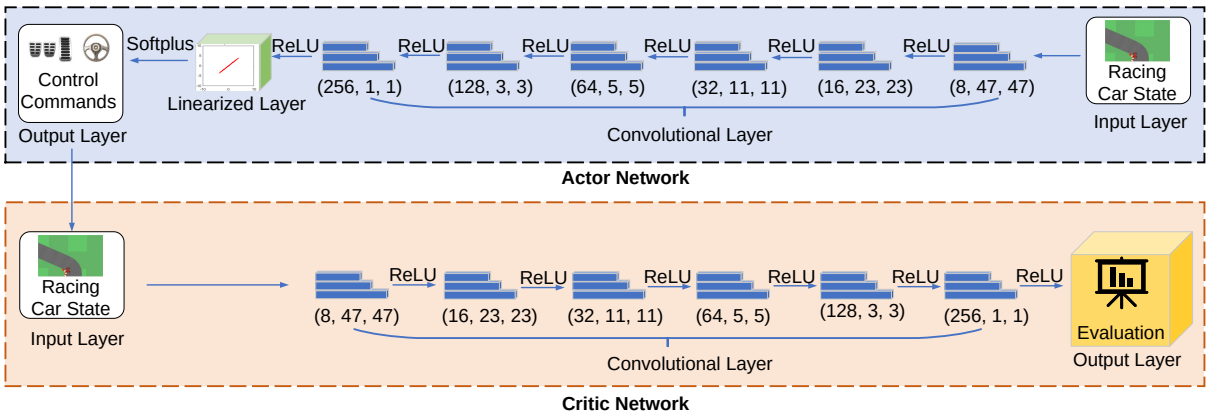


Figure 4.3: Structure of the image-efficient actor-critic network.

4.3 Decision Network

The decision network is to generate safe and efficient control commands during training. The decision network consists of two sets of image-efficient actor-critic networks that receive the sequence of images, the balanced rewards of actions and the curiosity reward respectively. The control policy in the actor-critic network compares the candidate control commands and chooses the best one based on their relative advantages.

4.3.1 Network Structure

The aforementioned two actor-critic networks select actions based on the states of the racing car. Given the proven effectiveness of convolutional neural networks (CNN) in image classification [237], a series of convolutional layers are used to extract essential information from raw image data. The control policy selects commands to minimize collisions and laptime. Fig. 4.3 illustrates the actor-critic network structure. While the current image-efficient actor-critic network effectively processes 2D image inputs, future advancements in autonomous driving simulations are expected to incorporate 3D image data for more realistic environmental perception. To enhance the network's ability to handle such inputs, integrating transformer-based architectures, such as Vision Transformers or Swin Transformers, would be a promising direction. These models can improve feature extraction and decision-making by capturing long-range dependencies and complex spatial relationships, ensuring robust performance in advanced simulators.

Algorithm 4: Actor-Critic Network

Input: State S_{t+1}, reward r_t
Output: Policy π_θ evaluated by the Actor-Critic Network
Initialize actor and critic network weights randomly
for each racing sequence **do**
  for $m = 1$ **to** M **do**
    for $t = m$ **to** T **do**
      Run actor network (AN) to receive an action a_t using current policy π_θ
      Run critic network (CN) to compute rewards $R_t, \ldots, R_T$
      Compute the advantage A of action a_t
      $A = R_m + \gamma R_{m+1} + \cdots + \gamma^T R_{m+T}$
    end for
  end for
  Update π_θ based on the computed advantage A
end for

The actor-critic network comprises an actor network (AN) and a critic network (CN) in similar structures. The AN generates candidate control commands and the CN assesses their relative advantages. The AN consists of an input layer, convolutional layers, a linear layer, and an output layer. It processes the current state, extracts features, adds linearity for better representation learning, and generates control commands. The ReLU activation is used to introduce non-linearity. The CN is composed of an input layer, convolutional layers, and an output layer. It uses the AN output and current state as inputs, extracts features, and selects the best control commands based on their evaluation. The CN aims to reflect long-term advantages over a period T, comparing the performance of selected control commands with the average performance.

The convolution layer (CL) is expressed as:

$$CL = (A, B, C) \tag{4.1}$$

where A , B , and C indicate the number of input channels, the number of output channels, and the kernel size, respectively. The actor-critic network for racing is summarized in Algorithm 4.

To verify the image-efficient property of the proposed network, a comparison is made against the SqueezeNet [238]. The SqueezeNet is designed to achieve high accuracy with significantly fewer parameters and a smaller model size, making it theoretically suitable for processing imagingltraining data. However, when applied to the training in car racing, SqueezeNet achieves a peak reward of around 150, which is substantially lower than the peak reward of 900 obtained by the proposed network. This suggests that although SqueezeNet is efficient in parameters, it might not be as effective in handling the specific characteristics of racing images.The limited capacity and heavy reliance on 1×1 convolutions in SqueezeNet restrict its ability to capture intricate spatial relationships. Additionally, fine-grained details that are crucial for optimal performance in car racing may also be inadequately represented. The proposed network proves to be more successful in capturing spatial dependencies and making accurate decisions in the complex task of car racing.

4.3.2 Control Policy Update of the Decision Network

The control policy is determined by the weights of the neurons in the decision network. Therefore, the weights of the neurons should be adjusted to optimize the control policy. Fig. 4.4 shows an example of a learning process involving a single racing sequence. The autonomous racing car starts from the starting point with the maximum score s_m . During the racing, two types of losses including safety loss L_s and efficiency loss L_e are defined. L_s increases as the distance to the track boundaries decreases. L_e is a constant value until the car completes the racing. When the autonomous racing car reaches the finish point, a final score is calculated.

The final score s_f is formulated as

$$s_f = s_m - L_s - L_e \tag{4.2}$$

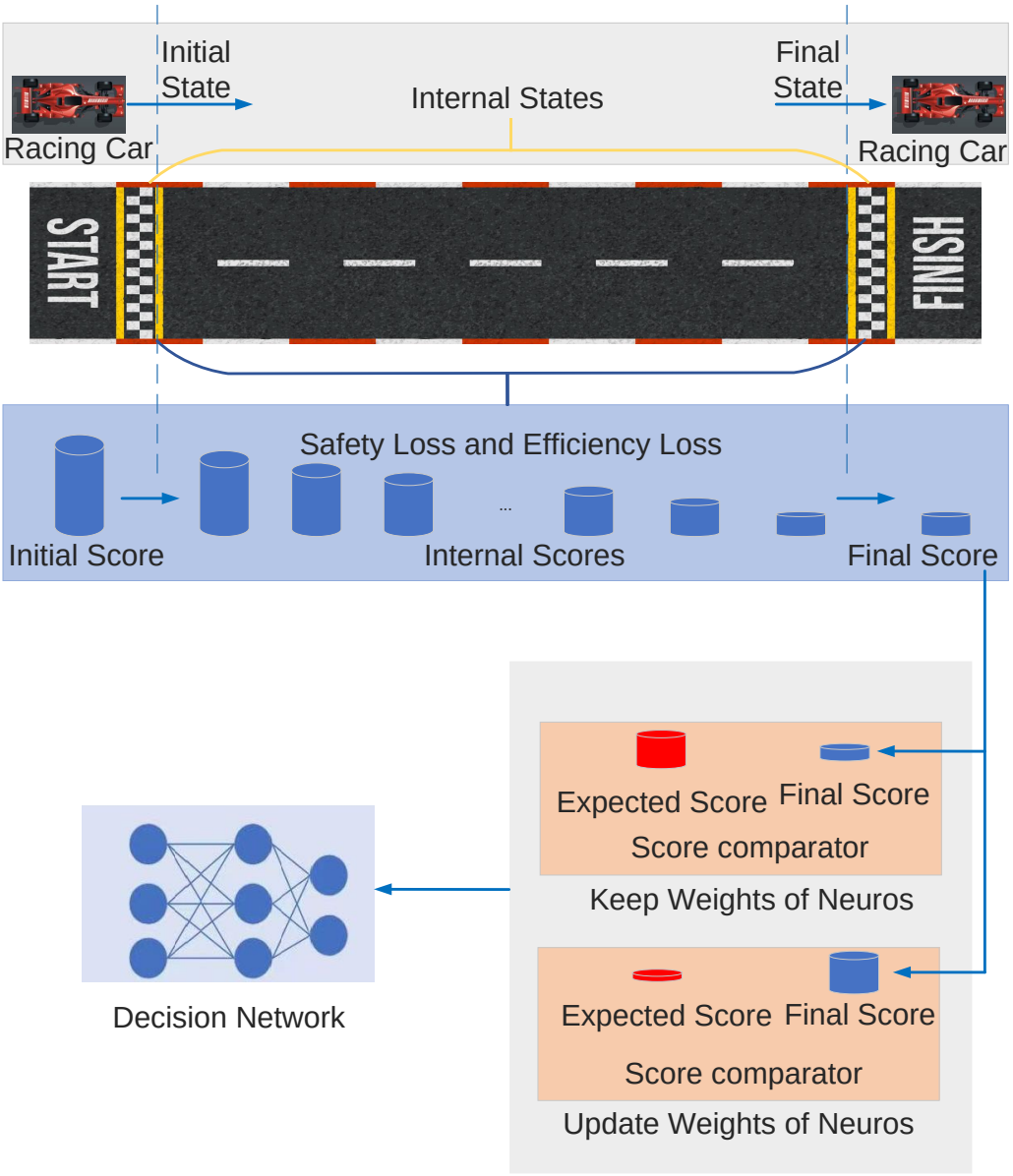


Figure 4.4: Control policy update of the decision network.

once the final score is obtained, a score comparator compares the final score with a predefined expected score. If the final score is higher than the expected value, the weights of the neurons in the decision network are updated. Otherwise, the weights are maintained, as the performance does not meet the expected level. When the racing car leaves the racetrack, the training score suffers significant safety losses, hindering the attainment of expected rewards. As decision sequences failing to reach the expected rewards are sieved out, the control policy updating prevents instances of the car veering off the track.

4.4 Curiosity-assisted Training Optimization

The curiosity-assisted optimization aims to enhance the training efficiency and the attention to dangerous sections, composing of the balanced reward function and the curiosity mechanism. The balanced reward function is to avoid collisions and reduce laptime during the training. The curiosity mechanism is to make optimal decisions in particular under hazard conditions.

4.4.1 Feature Encoding with CNNs

This thesis uses local perception. Therefore, the input to the curiosity mechanism consists of a sequence of raw images, $\{I_t\}_{t=1}^{T}$, captured from the racing environment over a period of time, from t to T. The I_t represents the image at time step t. To extract meaningful features from these images, the CNNs are employed as the feature encoder. The CNNs learn to detect local patterns and features in the input images, reduce the spatial dimensions and provide translation invariance. Let θ_f denote the parameters of the feature encoder. At each time step t, the CNNs process the input image I_t and output a feature vector $F_{m,t}$:

$$F_{m,t} = \mathrm{CNN}(I_t; \theta_f) \tag{4.3}$$

The encoded feature vector $F_{m,t}$ captures the relevant information from the input image I_t and serves as a compact representation of the racing environment at time step t.

4.4.2 Curiosity Mechanism

In DRL, the agent is expected to pay attention to specific sections of racetracks. However, the traditional agent explores each part of the game with equal attention, indicating that no particular areas receive highlighted emphasis. Although a high averaged reward generally signifies good performance, safety issues may persist in dangerous corners due to unequal focus. Hence, establishing an attention-distribution mechanism is necessary. To diversify the focus across distinct sections, the curiosity space S_c is denoted by

$$S_c = \mid F_m - F_p \mid \tag{4.4}$$

where F_p denotes the predicted encoded features. S_c quantifies the discrepancy between the outputs F_m and F_p. A higher value of the discrepancy indicates a poorer understanding of the environment. Therefore, this value enables the agent to identify sections of the racetrack where its understanding is lacking and that require further exploration. By encouraging targeted exploration in these sections, the agent can efficiently gather data and refine its understanding of the environment. This targeted exploration also helps maintain a balance between exploration and exploitation. Therefore, the agent is ensured not to get stuck in suboptimal behaviors and continuously improves its performance.

At each time step t, assume that the action a_t, current state s_t and next state s_{t+1} are known. The output encoded features of the current state $F_{m,s}$ and the next state $F_{m,s+1}$ could be obtained via feature quantifier vectors

$$F_{m,s} = q(s_t, \theta_f) \tag{4.5}$$

$$F_{m,s+1} = q(s_{t+1}, \theta_f) \tag{4.6}$$

where $F_{m,s}$ denotes the current encoded features. $F_{m,s}$ is taken as the input to obtain the predicted encoded features of the next state $F_{p,s}$

$$F_{p,s} = \mathrm{FM}(a_t, F_{m,s}) \tag{4.7}$$

where FM is the forward model to predict the feature representation of the next state. The curiosity reward space r_c could be obtained by

$$r_c = \beta \left\| F_{m,s+1} - F_{p,s} \right\|_2^2 \tag{4.8}$$

where β is a scaling factor obtained by calibration. The curiosity reward r_c plays a vital role in guiding the agent's exploration and enhancing its learning efficiency. While S_c quantifies the discrepancy between predicted and actual encoded features, r_c takes this discrepancy and directly incorporates it into the reward. The integration of curiosity into the reward function provides several advantages compared to using S_c alone. From a theoretical perspective, incorporating r_c directly into the reward modifies the DRL objec-

tive to include an intrinsic motivation component. This modification can be formalized by augmenting the traditional reward function with r_c as an integrated reward. By directly influencing the agent's reward, r_c helps prioritize actions that reduce significant uncertainty, leading to more efficient learning. The agent receives immediate feedback by exploring uncertain states, which is reflected in the integrated reward. The integrated reward encourages a balanced approach to exploration and exploitation. This balance is crucial in DRL, as it prevents the agent from focusing too much on curiosity (exploration) at the expense of task performance (exploitation).

The β parameter in the curiosity reward equation is used to appropriately scale the difference between the predicted feature vector $F_{p,s}$ and the real feature vector $F_{m,s+1}$, ensuring the difference contributes effectively to the training process. The value of β was determined through an adaptive tuning process. Initially, $\beta = 1$ was applied as a baseline. If the training performance with curiosity was worse than without it, β was increased to 2. If this adjustment improved performance, the value was further increased incrementally until a performance decline was observed. This approach ensured an optimal balance between exploration and task-specific learning.

By maintaining a constant exploration and learning, r_c helps the agent overcome unsatisfying sections associated with high discrepancy and facilitates continuous learning and improvement. Furthermore, β in the computation of r_c allows for the balancing of curiosity with the traditional reward. This ensures that the agent's exploration is flexibly guided by both curiosity and task-specific objectives.

4.4.3 Balanced Reward Function

The reward function is the feedback module that evaluates the actions generated by the decision network. During autonomous racing, the laptime and collision rates are the two major factors that evaluate the performance of the racing car. The laptime reflects the effectiveness of actions, while the collision frequency measures the safety of actions. Therefore, a good reward function for autonomous racing should guide the decision network to select actions that can avoid collisions with the track boundaries and reduce the laptime. However, the traditional reward functions assign equal attention to each step. The aver-

aged reward is heavily influenced by previous high-reward actions. Therefore, the averaged reward is not able to balance the historical and current rewards. The averaged reward function is defined as

$$r_{\text{ave}} = 0.99 r_{\text{ave}} + 0.01 r_{\text{current}} \tag{4.9}$$

where r_{ave} and r_{current} are the averaged reward for historical states and the reward of the current state, respectively. (4.9) represents an exponential moving average of the reward, commonly applied in PPO to maintain a stable performance estimate. Here, the weight 0.99 emphasizes historical rewards, ensuring stability by minimizing the impact of short-term fluctuations, while the 0.01 weight allows gradual adaptation to current rewards. This balance ensures consistent policy updates without overreacting to individual rewards, a standard practice in reinforcement learning to promote robust and adaptive training. The selected discount of 0.99 are used in [239], ensuring that the running reward estimate remains both reliable and informative throughout training.

Collisions during large and series bends at high speeds are the main safety concerns. The reward function should pay more attention to these critical steps, which are called corner rewards. However, the averaged reward cannot focus on dangerous scenarios effectively, as the averaged reward gives equal weights to all historical steps. Moreover, the longer the sequence length, the less attention it pays to the performance of each single step. To address this issue, a hyper parameter is introduced to balance the average reward and the corner rewards. With the hyper parameter, a balanced reward function is proposed to consider both the historical and current rewards

$$r_b = (1 - \gamma * N_c) r_{\text{ave}} + \gamma * N_c * r_{\text{current}} \tag{4.10}$$

where γ represents a hyper parameter that directs the racing car to prioritize random corners. r_b is the balanced reward under the current state of the racing car. N_c is the number of corners. If there are lots of corners, the current decision is more crucial and thus the discount of historical reward becomes higher. r_{ave} is formulated as

$$r_{\text{ave}} = \sum_{i=1}^{N-1} s_f^i \tag{4.11}$$

where N is the number of current step. s_f^i is the final score at i^{th} step. r_{current} is equal to s_f at N step. Therefore, γ promotes a safety-aware and forward-looking strategy, allowing the car to pay attention to possible dangers.

Constraints on the learning speed are also required to be limited within a fair range during each update. To improve the stability in learning, a clipped surrogate objective is used to control the learning speed. The clipped surrogate objective prevents significant adjustments of neurons that might lead to control policy divergence. The clipped surrogate objective is employed to update the policy network. The clipped surrogate objective is defined as

$$L_{\text{clip}} = \min(R * A, \text{clip}(R, 1-\sigma, 1+\sigma) * A) \tag{4.12}$$

where R is the ratio of the new policy probability to the old policy probability, and the clip() function ensures that each component of the gradient is limited between $1-\sigma$ and $1+\sigma$. A is obtained from the decision networks, and σ is a self-defined hyper-parameter constraining the change for the weights of neuros during each iterative update.

In this work, the σ hyperparameter was fine-tuned based on training observations. Initially set to 0.2, the standard value for PPO, σ was intended to facilitate moderate policy updates while preserving stability. However, during training, significant fluctuations in the reward curve were observed, leading to unstable learning characterized by high variance and inconsistent convergence across training episodes. This instability suggested that policy updates were overly aggressive, causing the agent to overfit to recent experiences while compromising its generalization across diverse scenarios. To address this issue, σ was gradually reduced to 0.1, ensuring that the R closer to 1, thereby encouraging more conservative updates. This reduction effectively constrained policy changes, preventing abrupt shifts in behavior between training iterations. As a result, the training process exhibited smoother convergence, with the policy improving consistently without pronounced performance fluctuations.

Algorithm 5: Curiosity-assisted Control Policy Update

Input: State s_0, hyper-parameter α
Output: Policy π_θ evaluated by the Critic-Actor Network
Randomly initialize Actor Network (AN), Critic Network (CN), Forward Network (FN), and Inverse Model (IM)
Initialize state s_0
Define the value of hyper-parameter α for FN and BN
for $m = 1$ **to** M **do**
    Use AN to obtain s_m, r_m, a_m, and s_{m+1}
    Compute curiosity reward r_c using Equation (6)
    Reconstruct reward $r_m \leftarrow r_m + r_c$
    Store s_m, r_m, a_m, s_{m+1} into the game sequence
    Compute advantage A using Algorithm 4
    Compute the loss of FN L_F by

$$L_F = \left\| F_{m,s+1} - F_{p,s} \right\|_2^2$$

    Compute predicted action a_p by IM

$$a_p = \mathrm{IM}(F_{m,s}, F_{m,s+1})$$

    Compute the loss of IM L_B by

$$L_B = \left\| a_p - a_t \right\|_2^2$$

    Update BN and FN using

$$\min_{\mathrm{AN,FN,BN}} (1-\alpha) L_B + \alpha L_F$$

    Update π_θ in AN and CN using Algorithm 4
end for

4.4.4 Curiosity-assisted Control Policy Optimization

The update of control policy based on the curiosity mechanism is summarized in Algorithm 5. The IM is a component of the decision network that predicts the actions based on the current state and a target state. The FN is the forward network that implements the forward model using a neural network. BN is the backward network that learns the rationale of selecting actions from a target state and moving backward to the current state. Unlike traditional control policy updates, the curiosity-assisted control policy updates both the FN and the BN. The update of the FN and the BN is conducted by balancing

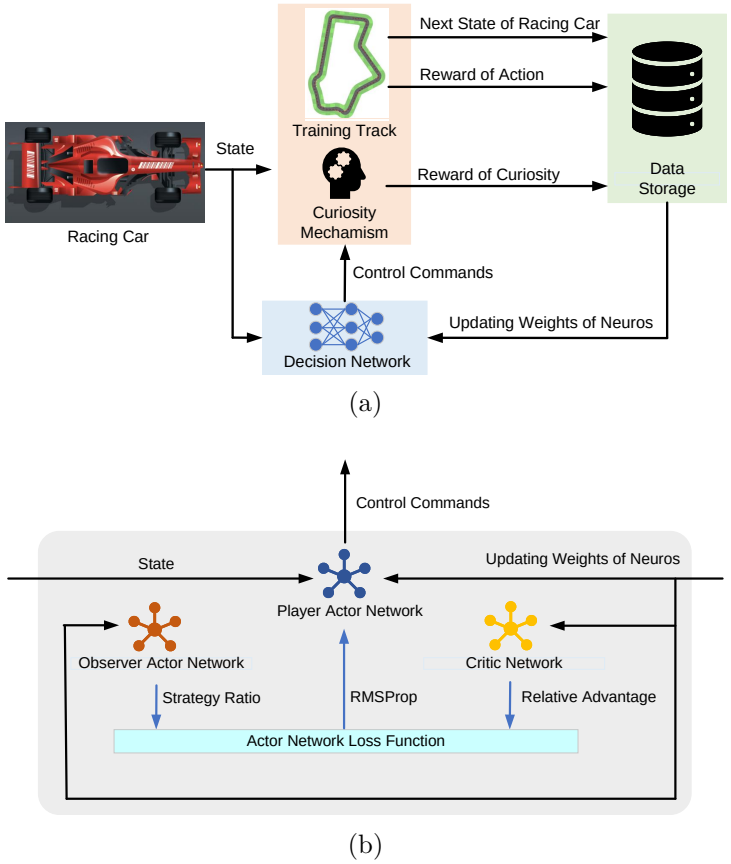


Figure 4.5: Curiosity-assisted control policy update of the decision network. (a) General process of the control policy update. (b) Internal structure of the decision network.

the losses of the BN and the FN using a scaling factor α. The FN generates the curiosity reward, and the BN explores the sections that need high attention. Fig. 4.5(a) illustrates the curiosity-based policy update. The generated data is stored in the data storage, which is the profit under the chosen actions based on the given state. The weights of the decision network are updated using both actions and curiosity rewards.

Fig. 4.5(b) illustrates the internal structure of the decision network. The observer actor network compares the ratio between the updated strategy and the previous strategy to measure whether the update is proper. The critic network provides the relative advantages to access the values of control commands. *RMSProp* is a process that helps train neural networks by adjusting the learning rate for each parameter.

The learning begins with an initial exploration. During the initial exploration, the agent randomly explores the racing environment. Therefore, the initial exploration allows the agent to form a preliminary understanding of the environment. After the initial exploration, the optimization is utilized to update the control policy network. With a contin-

uous interaction with the environment, the agent has the potential to focus on critical areas. The curiosity mechanism enables the focused learning by directing the attention towards regions with higher potential for improvement. Throughout the learning, the decision results are evaluated against a set of predefined metrics, such as the laptime and collision occurrence frequency. By evaluating the driving performance during the training, the algorithm can be fine-tuned for various car racetracks.

4.4.5 Curiosity-based Training with Balanced Reward Function

To incorporate both the balanced reward r_b and the curiosity reward r_c, a dual decision network is employed. The network consists of two separate networks: the primary decision network D_p and the curiosity decision network D_c. Out of which, D_p is trained using the balanced reward r_b, which reflects the agent's performance in terms of safety and efficiency. D_c is trained using the curiosity reward r_c, which encourages exploration based on the discrepancy between predicted and actual encoded features. At each time step t, the input image I_t is processed by the CNN feature encoder to obtain the encoded features $F_{m,t}$. These encoded features are then given to both decision networks in generating their respective actions $a_{p,t}$ and $a_{c,t}$:

$$a_{p,t} = D_p(F_{m,t}; \theta_p) \tag{4.13}$$

$$a_{c,t} = D_c(F_{m,t}; \theta_c) \tag{4.14}$$

where θ_p and θ_c denote the parameters of D_p and D_c, respectively. The action a_t executed in the environment is determined by D_p. The training for the dual decision network is illustrated in Algorithm 6. During each training episode, the racing environment is reset, and the initial state s_0 is obtained. At each time step t, the input image I_t is processed by the CNN feature encoder to obtain $F_{m,t}$. $F_{m,t}$ are then given to D_p and D_c in generating their respective actions $a_{p,t}$ and $a_{c,t}$. The action a_t executed in the environment is determined by D_p. D_p is updated using the tuple $(s_t, a_{p,t}, r_{b,t}, s_{t+1})$, which includes the balanced reward $r_{b,t}$. D_c is updated using the tuple $(s_t, a_{c,t}, r_{c,t}, s_{t+1})$, which includes the curiosity reward $r_{c,t}$. At the end of each training epoch, the weights of D_p and D_c are exchanged. This weight exchange allows the networks to share their learned knowledge and benefit from each other's experiences.

Algorithm 6: Curiosity-Driven Exploration with Balanced Reward Function

Input: Total epochs N, total episodes M, total timesteps T, hyper-parameter α, β
Output: Trained policy networks D_p and D_c
Randomly initialize CNN feature encoder with θ_f, D_p, D_c, Forward Model (FM), and Inverse Model (IM)
Initialize state s_0
Define the value of hyper-parameter α for balancing the losses
for epoch $= 1$ **to** N **do**
  for episode $= 1$ **to** M **do**
    Reset racing environment and obtain initial state s_0
    for $t = 0$ **to** $T-1$ **do**
      Obtain input image I_t from state s_t
      Compute encoded features $F_{m,t} = \text{CNN}(I_t; \theta_f)$
      Generate actions $a_{p,t} = D_p(F_{m,t}; \theta_p)$ and $a_{c,t} = D_c(F_{m,t}; \theta_c)$
      Execute action $a_t = a_{p,t}$ and observe next state s_{t+1}, reward $r_{e,t}$, and curiosity reward $r_{c,t}$
      Obtain input image I_{t+1} from state s_{t+1}
      Compute encoded features $F_{m,t+1} = \text{CNN}(I_{t+1}; \theta_f)$
      Predict encoded features $F_{p,t} = \text{FM}(F_{m,t}, a_t; \theta_{fm})$
      Compute curiosity reward $r_{c,t} = \beta \| F_{m,t+1} - F_{p,t} \|_2^2$
      Reconstruct reward $r_t = r_{e,t} + r_{c,t}$
      Store transition (s_t, a_t, r_t, s_{t+1}) into the replay buffer
      Compute the loss of FM $L_F = \| F_{m,t+1} - F_{p,t} \|_2^2$
      Predict action $a_{p,t} = \text{IM}(F_{m,t}, F_{m,t+1}; \theta_{im})$
      Compute the loss of IM $L_I = \| a_{p,t} - a_t \|_2^2$
      Update FM and IM by minimizing $(1-\alpha)L_I + \alpha L_F$
      Update D_p and D_c using the stored experiences in the replay buffer
    end for
    Exchange weights between D_p and D_c
    $\theta_p \leftarrow \theta_c$
    $\theta_c \leftarrow \theta_p$
  end for
end for

4.5 Real-time Proximal Control Policy Update

The real-time control policy update comprises the gradient-based control policy mechanism and the experience network. The experience network uses the gradient-based mechanism to adjust parameters and produce safe control commands.

4.5.1 Gradient-based Policy Update for Real-time Control

The PPO-C aims to learn the rules of choosing actions based on the states of the car and the local racing environment. Therefore, learning a precise policy in a short time is essential for effective DRL. The gradient policy method is applied to learn the control policy more efficiently, enabling the experience network to update its driving strategy by leveraging the gradient of rewards.

A racing sequence includes a series of states. Denote N as the number of separated states. The control commands are generated by a probabilistic network for each state. The probability of choosing a proper action in a given state is written as $p_\theta(a_t|s_t)$. θ represents the parameters of the policy model. The training involves continuously updating the probabilities of different control commands. The racing sequential probability is formulated as

$$S = (p_\theta(a_1|s_1), p_\theta(a_2|s_2), ..., p_\theta(a_N|s_N)) \tag{4.15}$$

During the racing, the number of collisions with the track boundaries indicates the level of safety. The laptime indicates the level of racing capability. The probability of achieving a game sequence p_m in the m^{th} track is defined as

$$p_m = \sum_{t=1}^{N} p_\theta(a_t|s_t) \tag{4.16}$$

Assume the total number of racetracks is M, and R_m is the reward among the m^{th} track. Thus, to generate a probability distribution suitable for safe and efficient driving on different racetracks, a reward function is defined as

$$R_{\text{total}} = \sum_{m}^{M} R_m p_m \tag{4.17}$$

where R_{total} is the total reward among M tracks. As R_m is a fixed value for the sequence m, p_m should be adjusted to increase R_{total}. The radient descent method is an effective way to update the decision network towards the desired outcomes. The desired outcomes are defined as shorter laptime and collision avoidance. The gradient descent method is

expressed as

$$\bigtriangledown f(x) = f(x) \bigtriangledown \ln f(x) \tag{4.18}$$

Lemma 1 is used to transform (4.18) to a more rigorous format.

Lemma 1. *For a differentiable function $f(x)$, the following equation holds:*

$$f(x)\frac{\mathrm{d}lnf(x)}{\mathrm{d}x} = f'(x)$$

Proof. Assume the initial conditions are

$$y = \ln f(x)$$

$$z = f(x) \tag{4.19}$$

Then we have

$$y = \ln z \tag{4.20}$$

According to (4.20), the following equation is obtained

$$\frac{\mathrm{d}y}{\mathrm{d}z} = \frac{1}{z} \tag{4.21}$$

From (4.19) we have

$$\frac{\mathrm{d}z}{\mathrm{d}x} = f'(x) \tag{4.22}$$

It is seen that

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{\mathrm{d}y}{\mathrm{d}z} \times \frac{\mathrm{d}z}{\mathrm{d}x} \tag{4.23}$$

According to (4.21), we have

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{1}{z} \times f'(x) \tag{4.24}$$

Afterwards, (4.24) could be further transferred to

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{1}{f_x} \times f'(x) \tag{4.25}$$

Combining $y = \ln f(x)$ with (4.25), the proof is finished with

$$f(x)\frac{\mathrm{d}\ln f(x)}{\mathrm{d}x} = f'(x) \tag{4.26}$$

□

With Lemma 1, (4.17) is further transferred to

$$\bigtriangledown R_{\text{total}} = \sum_{m}^{M} R_m p_m \bigtriangledown \ln p_m \tag{4.27}$$

Assuming that there is a long series of states, the probability of each racetrack p_m is extremely low and thus considered with the same small value. This small value is assumed to be in accordance with classical probability distribution, and (4.27) is further transferred to

$$\bigtriangledown R_{\text{total}} = \frac{1}{m}\sum_{m}^{M} R_m \bigtriangledown \ln p_m \tag{4.28}$$

Finally, each racing sequence could be expanded to N steps

$$\bigtriangledown R_{\text{total}} = \frac{1}{m}\sum_{m}^{M} R_m \sum_{t=1}^{N} \bigtriangledown \ln p_m(a_t|s_t) \tag{4.29}$$

The objective of (4.29) is to approach sequences associated with greater racing rewards. θ is updated by utilizing the rewards of each racing sequence. As illustrated in Fig. 4.6, the incorporation of a gradient-based mechanism significantly boosts the training efficiency by handling distinct states. Adhering to racing regulations, the racing reward is desired to reduce both laptime and achieve collision avoidance. The update process of θ among track m is formulated as

$$\theta = \theta + \alpha \bigtriangledown \log_e \pi_m(a_t|s_t) \tag{4.30}$$

where α is a parameter for the gradient exploration and π_m is the strategy network trained in the m^{th} track. The whole process is illustrated in Fig. 4.6

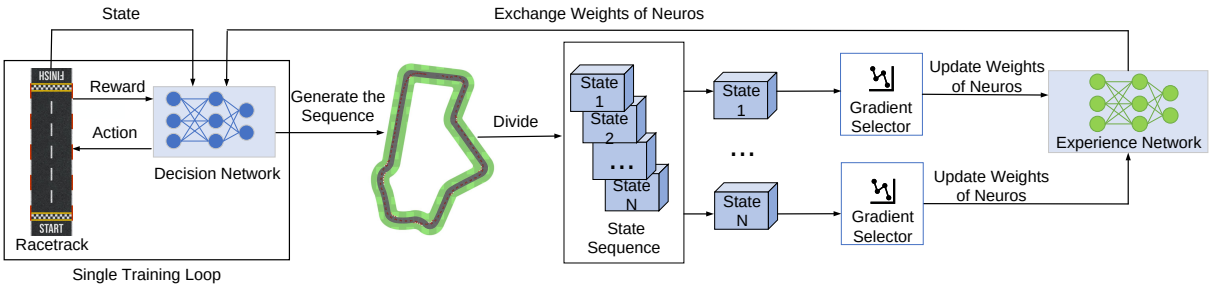


Figure 4.6: The gradient-based control policy update process.

4.5.2 Control Policy Optimization of the Experience Network

The experience network has the same actor-critic network as the decision network. Therefore, the experience network keeps updating the actor-critic network parameters until the average reward meets the desired value. The five steps to implement the control policy update of the experience network are below.

- The historical game sequences are stored in the experience repla, which is a recorder of the rewards for different combinations of states and actions. The current racing environment, the car state and the reward are used to update the experience replay.
- Candidate commands are generated according to the current state, racing environment and control constraints.
- The relative advantages of candidate commands are estimated for every state-action pair in the generated data through the actor-critic network.
- The optimal control commands are generated according to the relative advantages.
- The average reward over the past training is assessed. If the average reward is higher than the desired value, the experience network is updated using a gradient-based policy. Otherwise, the car state is updated and returned to the first step for new iterations.

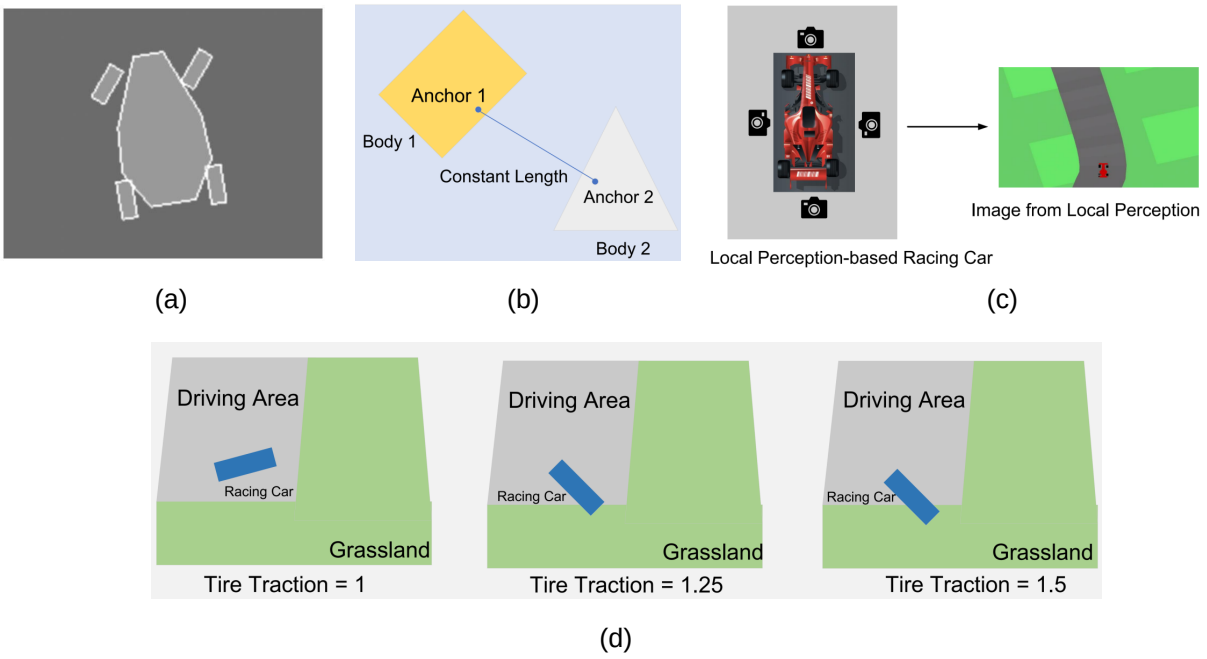


Figure 4.7: The physical rule-based racing setup and test racetracks in the Box2D. (a) The physical model of the racing car in the Box2D. (b) The definition of fixed distance in the Box2D. (c) The local perception of racing car in the Box2D. (d) The various tire traction in the Box2D.

4.6 Simulation Results

The simulations are designed to evaluate the safety and effectiveness of the PPO-C in different driving scenarios. To generalize the training results, racetracks are randomly selected from the candidate tracks. The training efficiency of PPO-C and three other DRL algorithms has been assessed in terms of training scores over various training epochs. The number of collisions with track boundaries and the lap time achieved by PPO-C have been compared and analyzed on 50 random racetracks. The racing performance at critical bends, trajectories, and variations of control levels are illustrated by four example cases.

4.6.1 Simulation Environmental Setup

The training and testing environment is Box2D, a widely used open-source physics engine designed to simulate and animate two-dimensional rigid-body dynamics [240]. In Box2D, the racing car is modeled as a rigid body with connected shapes, such as the chassis and wheels, resembling a real-world car. Figure 7(a) shows the car model in Box2D, which maintains a fixed distance between the body and tires, as exemplified in Fig. 4.7(b).

Box2D also supports local perception, with cameras capturing images for the decision network, as illustrated in Fig. 4.7(c). Additionally, Box2D realistically models tire traction and body damping, considering car-track interactions, as illustrated in Fig. 4.7(d). Tire traction varies with the contact area, and damping influences stability, simulating real-world conditions. Moreover, Box2D uses collision filtering to manage collisions between the car and track boundaries, enabling realistic suspension system simulations and enhancing simulation fidelity.

4.6.2 Results and Analysis

4.6.2.1 Car Dynamics

In order to reduce the computing burden of PPO-C, a bicycle model is used for the racing car in Box2D [241]

$$\dot{x} = V\cos(\varphi + \beta) \tag{4.31}$$

$$\dot{y} = V\sin(\varphi + \beta) \tag{4.32}$$

$$\dot{\varphi} = \frac{V}{l_r}\sin(\beta) \tag{4.33}$$

$$\dot{V} = a \tag{4.34}$$

$$\beta = \tan^{-1}(\frac{l_r}{l_f + l_r}\tan(\delta_f)) \tag{4.35}$$

where x and y are the coordinates of car's centre of mass. l_r is the length between the center of mass and car's rear axle. l_f is the length between the center of mass and car's front axle. β is the angle of the velocity with respect to the longitudinal axis of the car. ψ represents the yaw angle. a and δ_f are chosen as the inputs. a is the car longitudinal acceleration

$$a = F_{\text{throttle,max}} u_{\text{throttle}} / M \tag{4.36}$$

where $F_{\text{throttle,max}}$ and u_{throttle} are the maximum force of engine and the input level of throttle gate, respectively. M is the mass of the car. δ_f is the steering angle given by

$$\delta_f = \delta_{\max} u_{\text{steering}} \tag{4.37}$$

where $\delta_{\max}$ is the maximum angle of steering and u_{steering} is the input of steering level. Therefore, the states of the car can be changed by adjusting the inputs u_{steering} and u_{throttle}. The proposed algorithm is also applicable to the Ackerman model.

The Box2D simulator used in this study effectively captures variable track friction and system inaccuracies. Different friction coefficients are applied to various surfaces, such as racetracks and grasslands, as demonstrated in Figure 4.7(d). This allows the simulator to realistically reflect how the vehicle's handling changes when transitioning between high- and low-friction surfaces. The friction modeling ensures that the agent experiences realistic traction conditions, influencing acceleration, braking, and cornering performance. To address system inaccuracies, Box2D simulates control execution latency by introducing a slight delay between the agent's decision and the vehicle's response. This time-step-based simulation ensures that the agent learns to adapt its decision-making under realistic conditions, accounting for both environmental complexity and system-induced delays. These features provide a comprehensive simulation environment for testing autonomous driving algorithms under real-world-like conditions.

4.6.2.2 Scenario Description

During the training, the racing car starts at the initial point and the race is considered finished when it returns to the initial point. The car must avoid racetrack boundaries to ensure the safety, beginning with an initial speed of 0 and aiming to reach the final point as quickly as possible. this chapter designs scenarios with varying degrees of racing aggressiveness to evaluate performance across different driving habits. The effectiveness of PPO-C is evaluated every 50 episodes. The car drives approximately 80 steps, typically encountering at least 6 curvy sections per racetrack.

- **Scenario I:** The car faces irregular racetracks with multiple curvy sections, increasing the difficulty of avoiding collisions. A penalty of 1 for efficiency at each step represents normal driving.
- **Scenario II:** The car has a higher penalty for efficiency of 1.5, demanding quicker completion during training, representing aggressive driving. All other settings are the same as in Scenario I.

Additionally, different minibatch sizes of 10, 12, and 15 are used to validate the effectiveness of PPO-C. Consistent performance across various minibatch sizes demonstrates the robustness of the algorithm, indicating its effectiveness is not batch-size dependent, making the results more reliable.

4.6.2.3 Fast Convergence by using the Image-based Curiosity Mechanism

Simulations are demonstrated in the training curves of image-based PPO-C without balanced reward, standard image-based PPO, and numerical features-based PPO-C across different minibatch sizes and scenarios. The numerical inputs used as embedded features include position, steering, and throttle openings. Fig. 4.8(a)-(f) plot the average reward against the epoch, showcasing learning performance over time.

In Scenario I, Fig. 4.8(a) shows that with a minibatch size of 10, image-based PPO-C significantly outperforms both PPO and numerical features-based PPO-C, achieving higher average rewards more rapidly and maintaining superior performance throughout training. Similarly, Fig. 4.8(b) and Fig. 4.8(c) depict minibatch sizes of 12 and 15, respectively, where image-based PPO-C achieves higher rewards earlier and consistently outperforms both PPO and numerical features-based PPO-C. In Scenario II, Fig. 4.8(d) with a minibatch size of 10 shows image-based PPO-C maintaining its superior performance, with higher average rewards across epochs. Figure 8(e) and Fig. 4.8(f) with minibatch sizes of 12 and 15, respectively, demonstrate that image-based PPO-C still outperforms both PPO and numerical features-based PPO-C. The poor performance of the numerical features-based PPO-C is due to the limited capability of CNNs to process numerical data effectively. Additionally, the numerical data can not reflect the distance of the racing car from the grasslands, contributing to the poor training results.

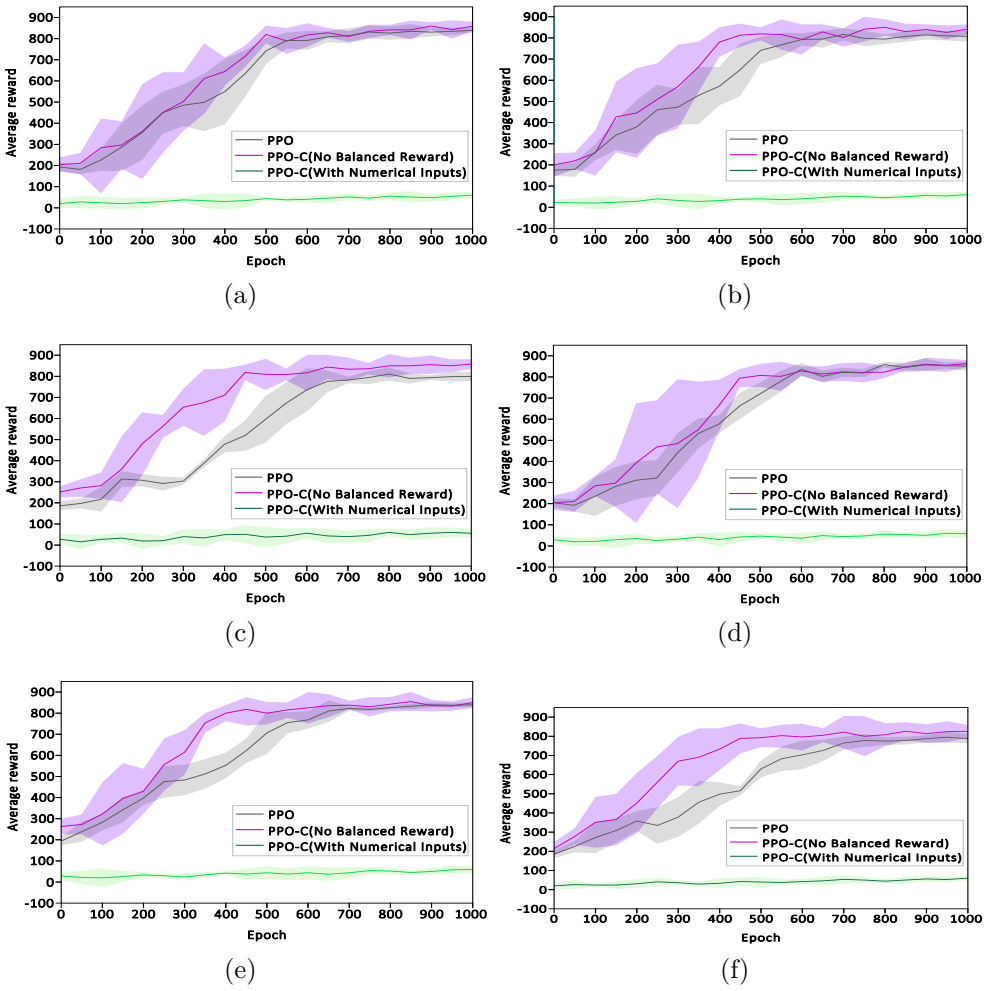


Figure 4.8: The training curves of PPO-C without balanced reward, normal PPO, and PPO-C with numerical inputs across different minibatch sizes and scenarios. (a) The average reward curve with a minibatch size of 10 in Scenario I. (b) The average reward curve with a minibatch size of 12 in Scenario I. (c) The average reward curve with a minibatch size of 15 in Scenario I. (d) The average reward curve with a minibatch size of 10 in Scenario II. (e) The average reward curve with a minibatch size of 12 in Scenario II. (f) The average reward curve with a minibatch size of 15 in Scenario II.

4.6.2.4 Reasoning Parameters of the Balanced Reward Function

Simulations are demonstrated in selecting the appropriate γ for a balanced reward function. To ensure sufficient and convincing simulations, it is assumed that the historical reward still constitutes the major portion of the total reward. Therefore, in this chapter, the minimum historical reward ratio is set at around 0.8. Considering that racetracks typically have approximately six corners in the Box2D environment, we select the maximum $\gamma = 0.04$:

$$\gamma = \frac{(1 - \text{historical reward})}{\text{number of corners}} = \frac{(1 - 0.8)}{6} = 0.036 \approx 0.04$$

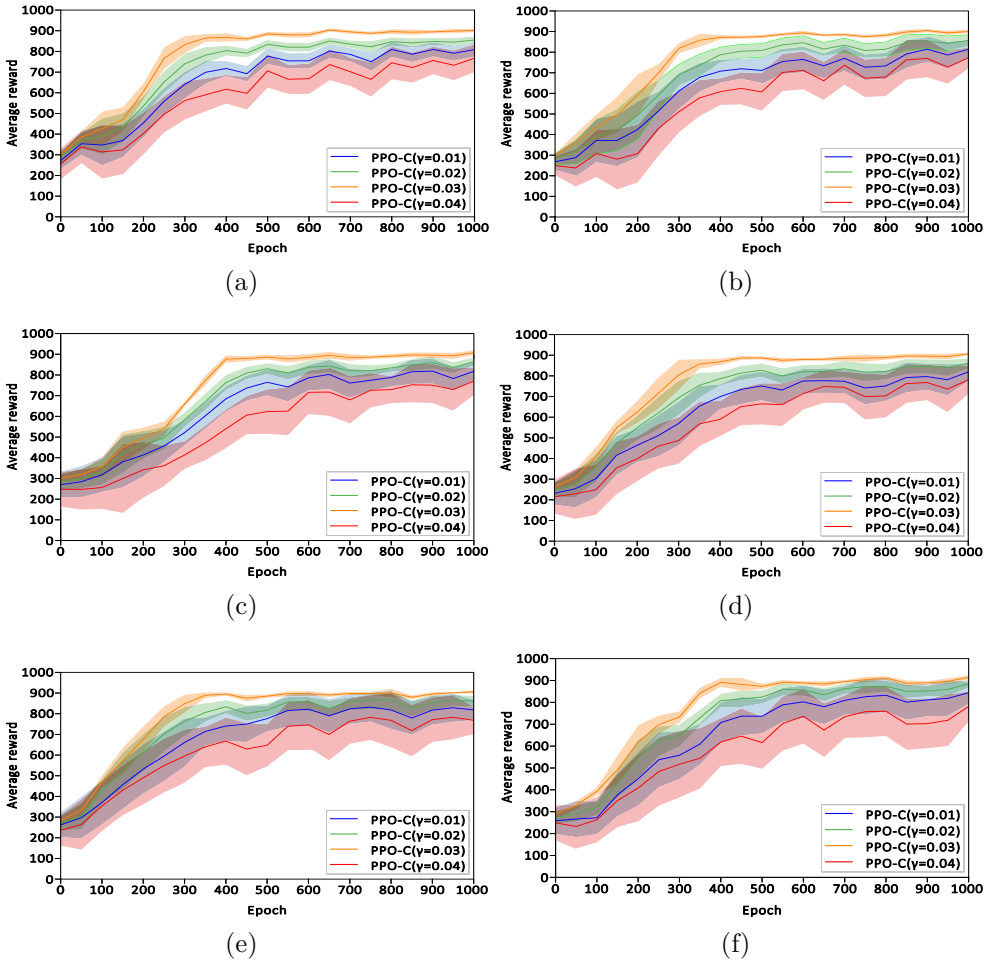


Figure 4.9: The training curves of the PPO-C with γ from 0.01 to 0.04 across different minibatch sizes and scenarios. (a) The average reward curve with a minibatch size of 10 in Scenario I. (b) The average reward curve with a minibatch size of 12 in Scenario I. (c) The average reward curve with a minibatch size of 15 in Scenario I. (d) The average reward curve with a minibatch size of 10 in Scenario II. (e) The average reward curve with a minibatch size of 12 in Scenario II. (f) The average reward curve with a minibatch size of 15 in Scenario II.

The other three candidate values for $\gamma = 0.01$, 0.02, and 0.03, respectively. To verify the generalization of the most suitable parameter for learning, three different minibatch sizes are used: 10, 12, and 15. Additionally, to confirm the adaptability of the best parameter across varied driving styles in racing, two different scenarios are employed to determine the most appropriate parameter.

Fig. 4.9 displays the training curves of PPO-C with various values of γ (ranging from 0.01 to 0.04) across different minibatch sizes and scenarios. Fig. 4.9(a) to Fig. 4.9(f) represent the following conditions: Fig. 4.9(a) to Fig. 4.9(c) are simulation results with minibatch sizes of 10, 12, and 15 in Scenario I, respectively; Fig. 4.9(d) to Fig. 4.9(f)

are simulation results with minibatch sizes of 10, 12, and 15 in Scenario II, respectively. Across all the test cases, there is a consistent trend of increasing average rewards with the number of epochs, generally stabilizing between 600 and 1000 epochs. Notably, the PPO-C with $\gamma = 0.03$ tends to perform better across multiple settings. Curves with $\gamma = 0.03$ consistently achieve higher average scores and show more stability as training progresses. For instance, in Fig. 4.9(a) and Fig. 4.9(d) with a minibatch size of 10, curves with $\gamma = 0.03$ demonstrate superior performance compared to other values. Similarly, in Fig. 4.9(b), Fig. 4.9(c), Fig. 4.9(e), and Fig. 4.9(f) with larger minibatch sizes, the curves with $\gamma = 0.03$ continue to outperform the others, achieving higher scores and smoother trends. The variability of the reward curves decreases with larger minibatch sizes, showing smoother trends for minibatch sizes of 15 compared to those of 10. Overall, $\gamma = 0.03$ is identified as the best-performing configuration across the various scenarios and minibatch sizes. Through comparisons with other benchmark algorithms, $\gamma = 0.03$ will be applied.

4.6.2.5 Comparison of Training Curves among Different Benchmark Algorithms

Fig. 4.10 displays the training curves of PPO-C compared with other benchmark algorithms across different minibatch sizes and scenarios. In Fig. 4.10(a) with a minibatch size of 10 in Scenario I, the PPO-C outperforms other algorithms consistently, achieving higher average scores and demonstrating more stability, especially noticeable after 400 epochs. In Fig. 4.10(b) with a minibatch size of 12 in Scenario I, the PPO-C shows superior performance, rising more sharply and stabilizing at a higher average score. Fig. 4.10(c) with a minibatch size of 15 in Scenario I shows the PPO-C continuing to outperform other algorithms, achieving higher average scores more quickly and maintaining steady improvement. In Scenario II, Fig. 4.10(d) with a minibatch size of 10, PPO-C remains the top performer, with its curve rising rapidly and stabilizing at a higher level. Fig. 4.10(e) with a minibatch size of 12 in Scenario II shows PPO-C outperforming SAC, PPO, and DDPG, achieving higher scores and showing less variability. Finally, in Fig. 4.10(f) with a minibatch size of 15 in Scenario II, the PPO-C maintains its lead, achieving

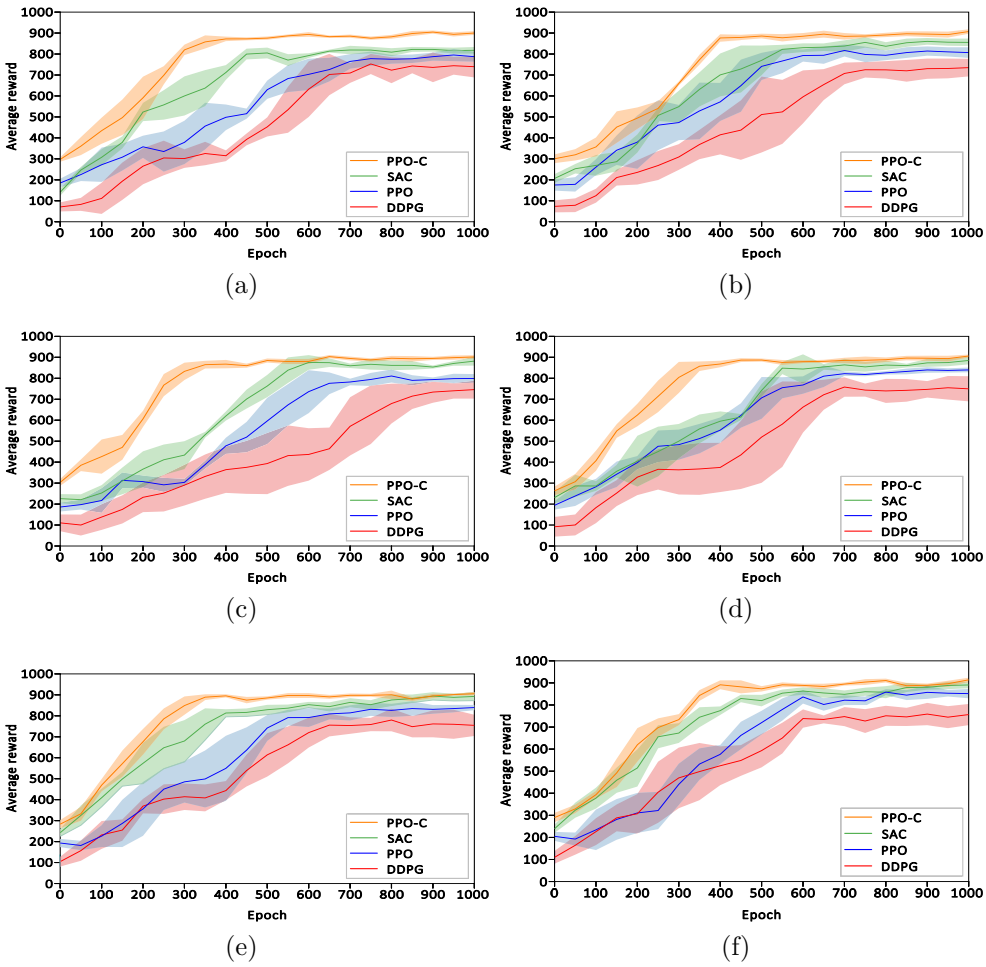


Figure 4.10: The training curves of PPO-C with other benchmark algorithms across different minibatch sizes and scenarios. (a) The average reward curve with a minibatch size of 10 in Scenario I. (b) The average reward curve with a minibatch size of 12 in Scenario I. (c) The average reward curve with a minibatch size of 15 in Scenario I. (d) The average reward curve with a minibatch size of 10 in Scenario II. (e) The average reward curve with a minibatch size of 12 in Scenario II. (f) The average reward curve with a minibatch size of 15 in Scenario II.

higher average scores and exhibiting smoother trends. Overall, the PPO-C consistently demonstrates superior performance across various scenarios and minibatch sizes, achieving higher average scores and showing more stability compared to SAC, PPO, and DDPG, underscoring its robustness and adaptability in different training conditions.

Table 4.2: Average Number of Collisions among 50 Racetracks

		Number of Collisions			
		PPO-C	**SAC**	**PPO**	**DDPG**
Scenario I	Minibatch 10	**0.54**	1.26	2.16	2.86
	Minibatch 12	**0.46**	0.82	1.72	2.76
	Minibatch 15	**0.42**	0.56	1.66	2.74
Scenario II	Minibatch 10	**0.64**	0.72	2.68	3.24
	Minibatch 12	**0.52**	0.62	2.46	3.16
	Minibatch 15	**0.48**	0.66	2.06	3.22

Table 4.3: Average Laptime among 50 racetracks

		Laptime			
		PPO-C	**SAC**	**PPO**	**DDPG**
Scenario I	Minibatch 10	**24.13**	26.23	26.73	25.32
	Minibatch 12	**23.52**	25.14	26.32	25.76
	Minibatch 15	**22.91**	24.36	25.96	24.74
Scenario II	Minibatch 10	**23.93**	24.12	25.33	24.76
	Minibatch 12	**22.26**	23.88	24.74	24.56
	Minibatch 15	**22.44**	22.56	23.77	24.25

4.6.2.6 Performance Analysis of Simulation Results Over Benchmark Algorithms

The PPO-C algorithm is compared against three benchmark algorithms recently used in racing, PPO, DDPG and SAC. Table 4.2 compares the laptime of the PPO-C and other benchmark algorithms among 50 random racetracks across different racing conditions and minibatch sizes. In normal racing, PPO-C records the minimum laptime of 24.13, 23.52, and 22.91 for Minibatch 10, 12, and 15 respectively. In Aggressive Racing, PPO-C continues to lead with minimum number of collisions of 23.93, 22.26, and 22.44 for the same minibatch sizes. SAC remains competitive, typically ranking second, while PPO and DDPG exhibit longer laptime. These results highlight PPO-C's superior capability in minimizing the laptime, demonstrating its effectiveness in both normal and aggressive racing scenarios.

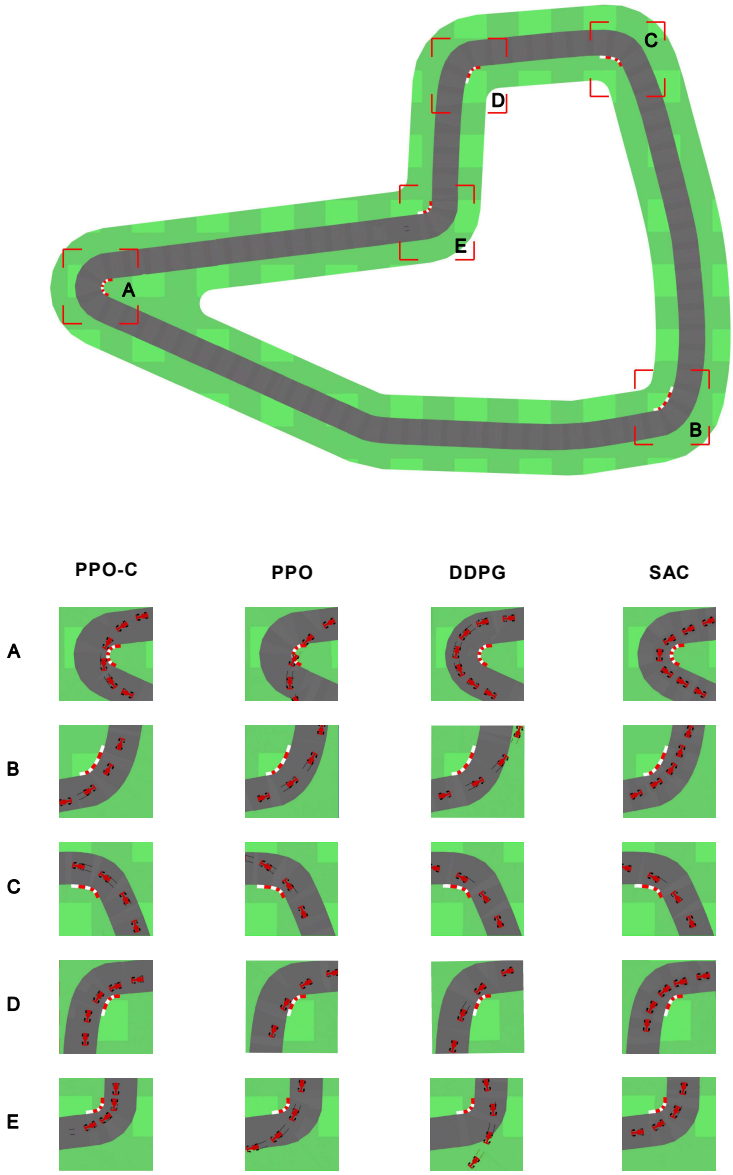


Figure 4.11: Driving performance of using PPO-C, PPO, DDPG and SAC in an Example Case.

Table 4.3 compares the number of collisions of the PPO-C and other benchmark algorithms among 50 racetracks across different racing conditions and minibatch sizes. PPO-C achieves a minimum number of collisions of 0.54, 0.46, and 0.42 for Minibatch 10, 12, and 15 respectively in normal racing, and 0.64, 0.52, and 0.48 in aggressive racing. SAC consistently ranks second in performance, followed by PPO and DDPG with higher collision rates. These results suggest that PPO-C excels in minimizing collisions across varying racing dynamics and minibatch sizes.

Table 4.4: Comparison of Average Speed and Average Lateral Acceleration in 5 Corners

	Metrics			
	PPO-C	**SAC**	**PPO**	**DDPG**
Average Speed (m/s)				
Corner A	**20.69**	19.23	17.23	18.86
Corner B	**24.45**	23.67	19.36	22.72
Corner C	30.67	32.98	26.46	28.23
Corner D	**20.87**	20.45	18.34	19.66
Corner E	**15.99**	15.56	14.65	9.43
Average Lateral Acceleration (m/s^2)				
Corner A	**3.95**	4.37	5.76	-2.23
Corner B	**6.32**	7.26	6.30	-4.22
Corner C	**8.57**	9.37	9.15	-0.28
Corner D	**1.25**	1.62	2.02	-5.32
Corner E	-2.62	-2.21	-5.91	-7.38

Fig. 4.11 illustrates how PPO-C and the other benchmark algorithms react to dangerous bends in an example case. There are five bends from *A* to *E* in this case. Bend *A* has a high curvature, making it challenging to drive through. Bends *B* and *C* are normal bends, requiring moderate control. Bends *D* and *E* are close to each other, increasing the difficulty of steering. It can be seen that PPO-C demonstrates safer and smarter driving than the other three algorithms, as it travels within the boundaries and stays close to the inner side of the curve when possible. In bend *A*, PPO deviates from the driving area, causing a high safety loss. DDPG follows the outer and middle side of the track, increasing its efficiency loss. SAC drives along the inner track, decreasing the time consumption. In bend *B*, DDPG also leaves the driving area, leading to a high safety loss. In bends *C* and *D*, PPO-C stays in the center of the track and drives along the track boundary, respectively, balancing the safety and efficiency objectives. DDPG and SAC move closer to the inner side of the track boundary, improving their efficiency performance. Bends *C* and *D* suggest that PPO-C is willing to sacrifice some efficiency profits to avoid collisions. In bend *E*, both PPO and DDPG exit the driving area, resulting in a high safety loss. In Table 4.4,

the comparison of average speed and average lateral acceleration across five corners for different DRL algorithms is illustrated. For average speed, PPO-C demonstrates higher levels in four out of five corners compared to SAC, PPO, and DDPG. This suggests that PPO-C adjusts its speed effectively on straight sections before entering corners, indicating a balanced approach that takes into account the connection between straight sections and corners. Higher speeds in straight sections can contribute to maintaining competitive performance while ensuring stability during cornering, as evidenced by PPO-C's consistent higher speeds.

Regarding average lateral acceleration, PPO-C generally exhibits lower acceleration levels in corners A to D compared to other algorithms. Lower lateral acceleration indicates smoother and more stable driving, reflecting the ability of PPO-C to make balanced decisions and maintain stability throughout the track. Notably, corners D and E, being closely positioned, highlight a strategy where acceleration is applied in the first corner and deceleration in the subsequent one, optimizing control and speed management through successive turns.

Conversely, DDPG shows lateral deceleration across most corners, implying potentially higher speeds on straight sections followed by necessary deceleration in corners to maintain control. However, the high lateral acceleration in corner E for DDPG suggests challenges in maintaining control within the track boundaries, leading to instances where the vehicle exceeds the driving area.

4.6.2.7 Driving Performance and Control Levels in Three Sample Cases

There are no shortcuts in the testing tracks, ensuring the algorithm cannot exploit any contingencies. The testing tracks feature sharp or multiple curves, increasing difficulty. The racing car starts from the center of the starting point and aims to reach the end point quickly. Figure 4.12 demonstrates that the racing car follows a safe and efficient trajectory within the feasible racetracks. Fig. 4.12(a) to Fig. 4.12(c) show the trajectories of Case 1 through Case 3, respectively, with a color bar indicating steering and throttle opening ranging from -1 to 1. Fig. 4.12(d) to Fig. 4.12(f) illustrate the steering angles

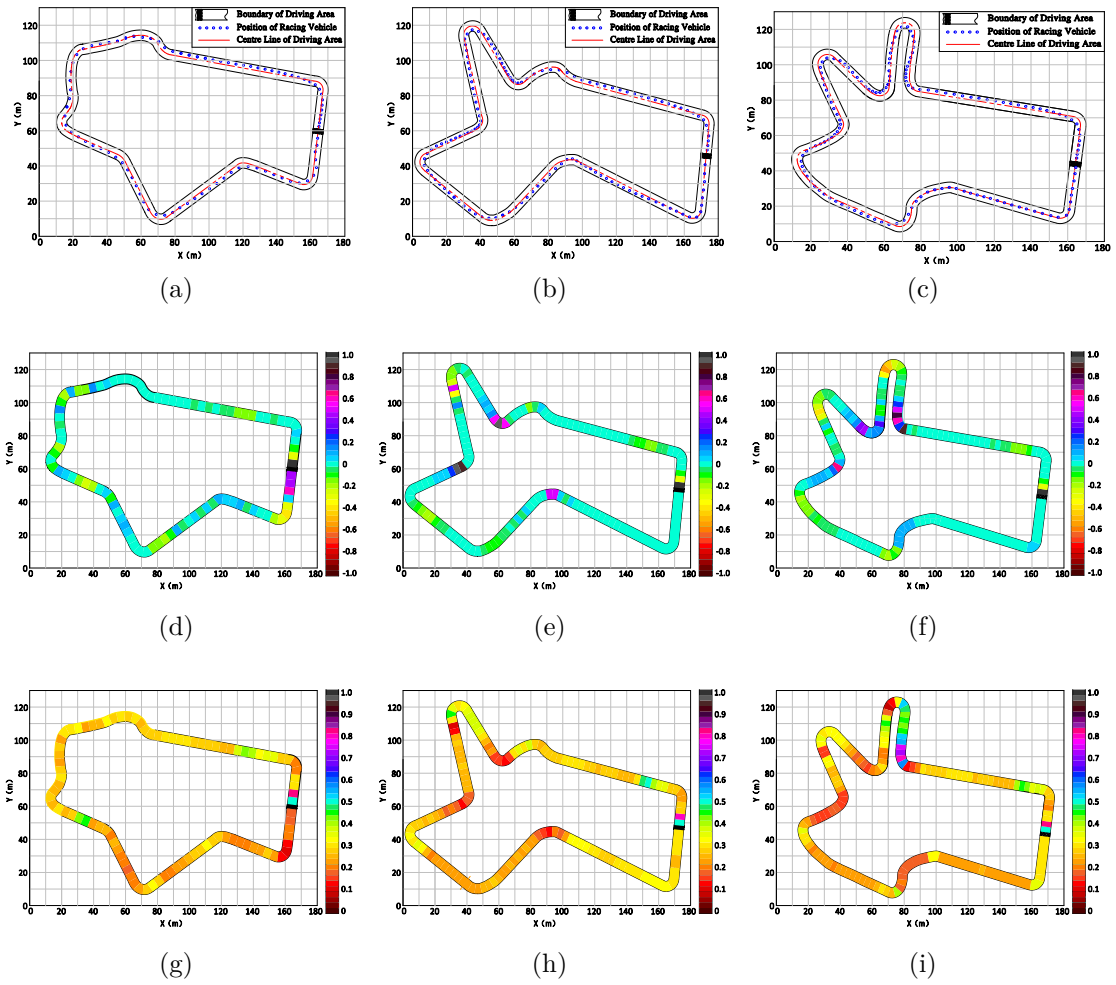


Figure 4.12: The driving performance and control levels of three sample autonomous racing cases. (a)-(c) show the trajectories of cases 1-3, respectively. (d)-(f) show the steering angles of cases 1-3, respectively. (g)-(i) show the throttle openings of cases 1-3, respectively.

of Case 1 through Case 3, respectively, and Fig. 4.12(g) to Fig. 4.12(i) show the throttle openings for these cases. In Fig. 4.12(a), the car deviates from the inner track boundary to avoid collisions. In Fig. 4.12(b) and Fig. 4.12(c), the car prefers the inner side of most curves to minimize lap time. These results show that PPO-C effectively balances safety and efficiency. Fig. 4.12(d) indicates that the car maintains its steering within -0.2 to 0.2 on curvy roads without large bends. In Fig. 4.12(e), the car exhibits both high steering around large bends and minor adjustments around consecutive bends. In Fig. 4.12(f), the car adjusts its steering angle more frequently due to larger and more consecutive bends, preferring slight steering on small bends and sharper steering on large bends. Fig. 4.12(g) illustrates that the car briefly increases its throttle opening when leaving curvy sections. In Fig. 4.12(h), the car reduces its throttle opening when passing the second bend in a

Table 4.5: Comparison against other learning-based methods

Methods	LPR	ITE	SL	RC	SMRT	VVCM
Salvaji et al. [242]	-	√	-	-	-	-
Spielberg et al. [243]	-	√	√	-	√	√
Evans et al. [244]	-	√	√	√	√	√
Ghignone et al. [245]	√	-	√	-	√	-
Proposed	√	√	√	√	√	√

Abbreviations: LPR: Local perception-based race; ITE: Improved training efficiency; SL: Shorter laptime; RC: Reduced collisions; SMRT: Simulation with multiple racetracks; VVCM: Visible variation of control commands; -: not considered or not given.

series. In Fig. 4.12(i), the car changes its throttle more frequently due to larger and more consecutive bends, maintaining a throttle opening around 0.3 on straight roads. Thus, the throttle control strategy involves steady acceleration on small bends and more pronounced adjustments for a series of bends.

To illustrate the advantages of the proposed algorithm, this chapter benchmarked against recent studies in Table 4.5. The DRL in [242] demonstrates enhanced training efficiency but overlooks other key factors, including reducing laptime, fewer collisions, validating performance across multiple tracks, and providing visualizations of control commands. On the other hand, [243] introduces a DRL that encompasses improved training efficiency, shorter laptime, validation across various tracks, and clear visualization of control commands. However, it overlooks the aspect of reducing collisions. In contrast, the algorithms proposed in [244] considered all the factors in both [242] and [243], but still heavily relies on global perception. Furthermore, [245] focuses solely on local perception, emphasizing shorter laptime and validation across various tracks. However, [245] neglects improvements in training efficiency, collision reduction, and variations in control commands.

The proposed algorithm reduces dependency on sophisticated equipment and achieves enhanced training efficiency. Moreover, the laptime is reduced and collisions are avoided, thereby the overall racing performance is improved. Furthermore, validations on multiple tracks have been made, while interpretable control commands are provided, showcasing the generalization and interpretability of the proposed algorithm.

4.7 Evaluation Based on DDTUI

The PPO-C has considered and verified all the factors of DDTUI, contributing to state-of-the-art DRL-based decision-making for autonomous driving. For driving safety, PPO-C employs a substantial collision penalty to ensure that the autonomous car avoids collisions with track boundaries. Verification is achieved by comparing the collisions of PPO-C with those of other benchmark algorithms, indicating fewer collisions. Regarding driving efficiency, PPO-C uses an efficiency penalty to encourage the AV to reach the endpoint as quickly as possible. Verification is achieved by comparing the lap times of PPO-C with those of other benchmark algorithms, demonstrating shorter lap times. For training efficiency, PPO-C utilizes a curiosity mechanism to facilitate faster convergence, verified by comparing its convergence rates with those of other benchmark algorithms, which indicate faster convergence.

The curiosity mechanism fits well with the idea of intrinsic interpretability, as it inherently provides a rationale for an agent's actions that is understandable to humans [246, 247]. Curiosity makes the decision-making process more self-explanatory by focusing the agent's exploration efforts on reducing uncertainty or discovering novel situations, aligning with the principle of designing interpretable models by restricting unnecessary complexity and providing clear motivations. Environmental consideration is addressed by penalizing collisions with track boundaries, which minimizes damage to surrounding grasslands and protects the environment. Verification is illustrated through the AV's trajectories, which remain within designated driving areas without veering off-track. PPO-C takes into account the needs of users, companies, and the public environment, supporting the adaptation of DRL-based decision-making as a practical real-world solution.

4.8 Discussion

The PPO-C algorithm typically surpasses comparative benchmarks by achieving greater training efficiency, higher average rewards, collision avoidance, and reduced laptime. Notably, while PPO-C approaches the highest training scores, it remains approximately 100 points behind, indicating the room for improvement. Future developments aim to narrow

this gap, ideally to within 50 points of the top score. Although the PPO-C demonstrates proficiency in static environments, its performance in dynamic settings requires further validation. Additionally, there is potential to decrease laptime, as the PPO-C has not yet completely optimized for inner track navigation, as shown in Fig. 11. Before real-world application, the PPO-C's policy network and reward function must undergo refinement and rigorous testing to ensure safety and reliability. Moreover, prior to real-world implementation, a higher-fidelity simulation environment will be utilized to bridge the gap between simulation and actual conditions effectively.

4.9 Summary

This chapter proposed a local perception-based, image-efficient, and balanced reward-orientated PPO-C for autonomous racing. The PPO-C aims to improve the training efficiency and driving performance of the racing car. To enhance the attention to critic steps, a balanced reward function is used to balance the historical and current rewards during the training. To enhance safety in exploration, a curiosity mechanism is introduced to focus on the dangerous racing periods. The results demonstrate that as training time increased, the proposed PPO-C improves its average scores with a higher degree of safety. Comparisons among the PPO-C and other three representative DRL algorithms were conducted, showing that the proposed algorithm outperforms in terms of no collision, shorter laptime, shorter training time, and higher average rewards. In the future, extensive research will be conducted in several aspects, including 1) verifying the racing ability of PPO-C under more uncertain conditions, and 2) optimizing the racing process considering diverse objectives such as riding comfort.

Chapter 5

Conclusion

This chapter summarizes the contributions of the current work. Furthermore, this chapter highlights the present work's limitations and potential future work to overcome these limitations.

5.1 Research Contributions

This work has explored DRL-based decision-making for autonomous driving in detail. It proposes a rationale evaluation framework for DRL-based decision-making algorithms, a holistic architecture that enables highway driving in accordance with the proposed evaluation framework, and an integrated decision-making algorithm for autonomous racing that considers this framework. Results for each research objective mentioned in previous chapters will be presented in order to highlight the contributions made by this work.

5.1.1 This Thesis Proposes a Rationale Evaluation Framework for DRL Decision-Making

A rationale evaluation framework is beneficial for the development of DRL-based decision-making and the transition from concept to real-world products. Currently, only general evaluation frameworks for AI products have been proposed, such as in [93]. Therefore, a rationale evaluation framework for DRL-based decision-making in autonomous driving should be designed. This thesis proposes a rationale evaluation framework for DRL-based decision-making in autonomous driving, focusing on five key factors: driving safety, driving efficiency, training efficiency, unselfishness, and interpretability. While the framework has demonstrated effectiveness within simulation environments, further validation

in real-world scenarios remains necessary to confirm its practical applicability. Driving safety remains the foremost requirement, as minimizing collisions directly correlates with improved safety outcomes and adherence to rigorous standards [65, 66]. Similarly, driving efficiency not only enhances user experience but also optimizes road capacity, traffic flow, and emergency response capabilities, highlighting its multifaceted role in improving AV effectiveness and energy efficiency [74, 78]. Training efficiency is critical for AV development, as it directly impacts the time and computational resources required, ultimately reducing costs and device wear [36, 81]. Furthermore, unselfishness, or the AV's ability to account for the intentions of human-driven vehicles, fosters cooperative driving behaviors, promoting smoother, more harmonious road interactions and mitigating potential conflicts [82, 85]. Lastly, interpretability is essential for ensuring that DRL-based algorithms make transparent and justifiable decisions, thereby enhancing user trust and enabling compliance with regulatory standards [93, 101]. These factors create a robust framework for evaluating and improving DRL-based decision-making algorithms for autonomous driving, advancing the field toward safer, more efficient, and more user-centered AV systems.

5.1.2 Summarization for DRL Decision-Making Across Scenarios.

The different scenarios have various tasks and road features for driving, therefore the divided analysis and summary for each scenario is crucial for the future algorithm development. Given the DDTUI is proposed as a good evaluation framework, each scenario is evaluated based on DDTUI in this thesis. Through recent researches, efficiency is highly prioritized at intersections and ramps to optimize travel time and reduce congestion, cited in 94.7% and 87.5% of studies respectively. Safety is primarily emphasized on highways (76.7% of studies), where accident prevention is crucial at high speeds, whereas intersections receive less focus on safety. Training efficiency is highlighted at roundabouts (75%) and unsignalized intersections (68.4%) to support effective vehicle maneuvering in complex environments. Interpretability is notably valued at ramps (56.25%) and highways (36.7%), underscoring the importance of understandable decision-making. While

unselfishness is generally less emphasized, highways and ramps receive more focus in this area. The primary challenges identified for future DRL-based decision making algorithms include: (1) achieving a balance across all DDTUI factors in one unified framework, as very few studies address all five simultaneously, reflecting the complexity of integrated approaches; (2) improving interpretability without compromising model performance, as less than 40% of studies explicitly prioritize interpretability; (3) enhancing unselfish behavior in multi-agent scenarios through advanced techniques like integrating game theory with social value orientation; and (4) bridging the gap between simulation-based training and real-world deployment by developing more realistic simulation environments to ensure DRL models can adapt to real-world conditions. These challenges underscore the need for interdisciplinary approaches in advancing DRL-based autonomous driving and offer guidance for future research. By addressing these challenges, the field can move closer to achieving safe, efficient, and intelligent autonomous transportation systems.

5.1.3 Integrated DRL-Based Algorithm for Current DRL Shortcomings.

Recent advances in DRL have demonstrated state-of-the-art results in AV applications, particularly using methods like Q-Learning [30], DDPG [193], PPO [194], and DQN [190]. Each method brings unique strengths and limitations to AV decision-making. Q-Learning employs a state-action value function to determine optimal actions under various conditions [30] and is capable of selecting safe actions in constrained environments [31]. However, its application is limited to simpler tasks and scenarios due to its slow convergence speed [32]. DDPG, which utilizes deep neural networks for control policy approximation, has proven effective at improving convergence rates and driving performance in AV tasks [38, 39]. Nonetheless, DDPG faces challenges in exploring alternative actions and adapting to diverse driving conditions, as it tends to produce absolute results that limit flexibility in action selection. PPO, which employs a probabilistic control policy, offers improved exploration capabilities and has been successfully applied in multi-agent and high-density traffic scenarios [33, 34]. Despite these advantages, PPO often relies on generalized reward functions that dilute performance in lengthy training sequences, leading

to slower convergence in critical task-specific objectives [195]. DQN, a widely used DRL algorithm for AVs, excels in tasks involving discrete actions like acceleration, deceleration, and lane changes, and is computationally efficient due to its use of experience replay [35]. However, DQN struggles with complex scenarios requiring long-term dependencies and interpretability, which can lead to suboptimal decisions and unresolved collision-avoidance issues in interactive environments.

To address these limitations, this study proposes the RBDQN-CS (Risk-Balanced Deep Q-Network with Collision Supervision), an advanced DQN variant that introduces a risk-attention mechanism, balanced reward function, and collision-supervised mechanism. These enhancements are designed to improve interpretability, convergence efficiency, and safety in complex driving environments. Specifically, the risk-attention mechanism enables more nuanced attention to high-risk scenarios, the balanced reward function aligns reward distribution with critical objectives, and the collision-supervised mechanism enhances collision-avoidance capabilities. The effectiveness of RBDQN-CS is validated through performance evaluations on a three-lane highway, considering both normal (4-6 HDVs) and high (7-10 HDVs) traffic flows. In the first validation, RBDQN-CS is compared with baseline models—standard DQN, DQN with a balanced reward function (BDQN), DQN with a risk-attention mechanism (RDQN), and DQN with a collision-supervised mechanism (DQN-CS)—demonstrating superior performance in convergence speed, reward accumulation, collision rate reduction, and average speed maintenance. In a second validation, RBDQN-CS is benchmarked against established DRL algorithms, including DDPG [38], PPO [34], Advantage Actor-Critic (A2C) [191], and DQN [190]. RBDQN-CS consistently achieves higher rewards both at and after convergence, demonstrates a lower collision rate, maintains a higher average speed, and converges faster than these benchmarks in both normal and high-traffic scenarios. By addressing the challenges of computational efficiency, interpretability, and balanced exploration, RBDQN-CS represents a significant step forward in developing reliable, high-performing decision-making models for AVs in complex and dynamic environments, enhancing safety, efficiency, and robustness in autonomous driving systems.

5.1.4 Integrated DRL-based algorithm considering DDTUI simultaneously

As summarized in Section 5.1.2, addressing all five factors in DDTUI simultaneously is essential. The proposed RBDQN-CS addresses and validates each of these factors, advancing DRL-based decision-making toward state-of-the-art performance in autonomous vehicles. For driving safety, RBDQN-CS integrates a collision penalty mechanism that enables the AV to recognize and avoid potential collision scenarios as it converges. This penalty functions as a rule-based safety check, minimizing collision occurrences and ensuring that safety-related decisions are made transparently. The model's safety verification is established by comparing collision rates between RBDQN-CS and other benchmark algorithms, with results showing that RBDQN-CS consistently achieves lower collision rates. This rule-based mechanism also enhances interpretability by providing clear mathematical formulations that offer insights into the AV's decision-making process.

In terms of driving efficiency, RBDQN-CS applies an efficiency-focused reward structure that encourages the AV to reach its destination in the shortest possible time. The model's efficiency is validated by evaluating average speed metrics and comparing them to those achieved by other benchmarks; RBDQN-CS demonstrates a measurable improvement in average speeds, highlighting its ability to optimize travel time. For training efficiency, RBDQN-CS introduces three main strategies: a risk-attention mechanism, a balanced reward function, and a collision-supervision component. These mechanisms contribute to faster model convergence by guiding the training process in a more focused and efficient manner. The effectiveness of this training efficiency strategy is confirmed through comparisons of convergence rates with other benchmarks, showing that RBDQN-CS achieves faster convergence, indicating a reduction in training time and resource demand.

To address unselfishness, RBDQN-CS limits the AV's lane-changing actions, minimizing disruptions to nearby vehicles and promoting smoother traffic flow. Verification of this factor is demonstrated through scenario-based assessments, illustrating that RBDQN-CS keeps lane changes to a minimum, moving only when necessary from the starting lane to the target lane. This approach reflects a balanced consideration of user experience,

corporate objectives, and broader public traffic safety. By addressing each DDTUI factor in a structured and practical manner, RBDQN-CS advances DRL-based decision-making, offering a viable and well-rounded solution adaptable to real-world autonomous driving challenges.

5.1.5 Integrated DRL-based algorithm for current DRL shortcomings at extreme situations

Autonomous racing is a specialized case of autonomous driving. Compared to standard autonomous driving, autonomous racing requires decision-making at the extreme situations, increasing its difficulty. However, fully exploring extreme-case driving benefits the development of autonomous driving as these scenarios reveal the upper capabilities of autonomous systems. State-of-the-art DRL techniques such as DDPG, SAC, and PPO have demonstrated promising applications in autonomous driving scenarios, each with unique strengths and limitations [208]. DDPG, as an off-policy algorithm, uses deep neural networks to develop a deterministic control policy, allowing it to handle high-dimensional data effectively. However, its deterministic approach limits exploration, often resulting in suboptimal decisions in dynamic environments where adaptive exploration is critical [38, 39]. SAC, on the other hand, employs a maximum entropy framework, promoting robust exploration and achieving higher average speeds in competitive environments like racetracks [231, 232]. However, SAC's undirected exploration can lead to excessive and inefficient exploration, slowing convergence in more complex scenarios. PPO, with its policy gradient approach, offers stability and a probabilistic policy structure that enables faster exploration than deterministic algorithms. Nevertheless, PPO's efficiency declines in long-sequence, high-variability tasks, such as racing, where it may fall into local optima and struggle to generalize across diverse states and actions [233, 235, 236].

To address these challenges, this paper proposes the PPO-C algorithm, which enhances PPO by introducing a curiosity-driven exploration mechanism and a balanced reward structure to improve training efficiency, adaptability, and performance in complex environments. The curiosity mechanism directs PPO-C's attention to high-uncertainty and less-explored regions, fostering adaptive exploration that dynamically adjusts as the racing

environment evolves. Additionally, the balanced reward function helps distribute rewards effectively across short segments of crucial racing sections, enabling PPO-C to prioritize safety and efficiency without sacrificing speed in critical maneuvers. These enhancements are particularly beneficial in racing contexts, where standard reward structures often lead to sparse or insufficiently informative feedback, hindering performance optimization. The proposed PPO-C model was validated through extensive comparisons with DDPG, SAC, and PPO across multiple racing scenarios. Results demonstrate that PPO-C achieves superior performance in minimizing lap time and collision rates while exhibiting higher convergence stability and improved training consistency [34, 190]. Specifically, PPO-C outperformed the benchmarks in maintaining a balanced approach to speed and safety, showing significantly higher average speeds on straight sections and stable lateral acceleration through corners. Its ability to dynamically adjust speed and trajectory in response to track complexities, particularly in high-curvature sections, underscores its effectiveness in managing challenging racing scenarios [190, 239]. Furthermore, PPO-C's robustness was verified across various minibatch sizes, demonstrating consistent performance improvements regardless of batch configuration, which supports its adaptability and reliability in real-world applications.

For further verification, PPO-C was evaluated on diverse racetrack configurations, where it consistently outperformed other algorithms in terms of average speed, cornering stability, and avoidance of boundary collisions [35, 196]. It showed a clear advantage in promoting a balanced exploration-exploitation strategy by dynamically adjusting its policy to maximize both standard and curiosity-based intrinsic rewards, effectively enhancing learning in critical sections [214]. Notably, PPO-C achieves these improvements with reduced dependency on sophisticated sensing equipment, making it a practical solution for broader applications. In addition to these performance benefits, PPO-C produces interpretable control commands that align well with human driving expectations, facilitating a better understanding of the algorithm's decision-making process [93].

5.1.6 Integrated DRL-based algorithm at extreme situations considering DDTUI simultaneously

As highlighted in Section 5.1.2, addressing all five factors in the DDTUI framework simultaneously is critical for developing comprehensive DRL-based decision-making systems. The proposed PPO-C algorithm successfully addresses and validates each of these factors, pushing DRL-based decision-making toward state-of-the-art performance in autonomous racing. By integrating each DDTUI component, PPO-C provides a robust framework for autonomous driving that balances safety, efficiency, environmental considerations, interpretability, and user needs. For driving safety, PPO-C incorporates a significant collision penalty that discourages the autonomous vehicle from making contact with track boundaries, thereby reducing the likelihood of accidents. This safety mechanism was verified by comparing PPO-C's collision rates with those of other benchmark algorithms, with PPO-C showing consistently fewer collisions, indicating enhanced safety under high-speed conditions.

In terms of driving efficiency, PPO-C applies an efficiency-oriented penalty designed to motivate the autonomous vehicle to complete laps in the shortest time possible. This approach was validated through lap time comparisons with other algorithms, demonstrating that PPO-C achieves shorter lap times, thus optimizing both speed and efficiency on complex racetracks. Training efficiency is another critical factor that PPO-C addresses through its integrated curiosity mechanism, which guides the algorithm to explore high-value sections of the track more effectively. This mechanism encourages faster convergence by focusing learning on critical areas rather than exploring uniformly. Comparative analysis of convergence rates against other benchmarks confirmed that PPO-C reaches optimal policies faster, thus reducing training time and computational resources needed for deployment. The curiosity mechanism also aligns well with the concept of intrinsic interpretability by inherently offering an understandable rationale for the agent's actions to humans [246, 247].

Environmental considerations are incorporated through penalties for boundary collisions, which serve to keep the vehicle within designated driving areas, thereby minimizing potential damage to surrounding grasslands and protecting the natural environment adjacent to the track. This feature was validated by observing PPO-C's driving trajectories, which consistently stayed within the designated track boundaries, avoiding off-track excursions and preserving the surrounding landscape. Finally, PPO-C is designed to consider the diverse needs of users, companies, and the public. Its balanced approach to safety, efficiency, and environmental respect reflects real-world concerns, making PPO-C a viable solution for DRL-based decision-making applications in practical autonomous driving scenarios. By aligning with DDTUI principles, PPO-C promotes adaptive, efficient, and responsible decision-making, establishing itself as an advanced and well-rounded tool in the autonomous racing field.

5.2 Limitations

5.2.1 Limitation of the 2D Simulator

A primary limitation of this study lies in the reliance on a 2D simulator for training and evaluation. The 2D environment, while offering a simplified and computationally efficient platform, lacks the complexity and fidelity of a 3D simulation environment like CARLA. In 2D simulators, key factors such as vehicle dynamics, lighting conditions, and environmental variations are simplified, resulting in a less realistic representation of real-world conditions. In contrast, a 3D simulator offers advanced physics and detailed visual data that allow for more sophisticated environmental interactions. For instance, CARLA incorporates vehicle dynamics modeling, which accounts for acceleration, braking, and cornering forces, producing a more accurate and nuanced interaction between the vehicle and its surroundings. Additionally, 3D simulators capture changing lighting conditions, varying weather patterns, and complex terrains, all of which impact autonomous driving performance and require advanced algorithmic adaptations. These real-world factors are critical for robust algorithm development, as they enable the model to learn and adapt

to dynamic scenarios that are likely to be encountered in actual driving situations. By transitioning to a 3D simulator, future work could enhance the depth of learning and provide a more comprehensive assessment of the algorithm's performance, thereby ensuring a smoother transition from simulation to real-world application.

5.2.2 Limited Verified Scenarios

Another limitation of the current study is the narrow scope of scenarios, specifically focusing on a controlled highway environment using the RBDQN-CS algorithm. While effective in this context, this approach does not encompass the variety of real-world driving situations encountered on public roads. Complex scenarios such as on-ramp merging, roundabouts, and unsignalized intersections introduce unique interaction dynamics and decision-making complexities that are distinct from highway driving.

On-ramp mergings require vehicles to adjust speed and safely enter the main highway. To handle this scenario, an DRL-based system must be able to predict the behavior of surrounding vehicles while accounting for factors like limited visibility and fluctuating speeds. A potential solution is to incorporate a mechanism for multi-agent interaction prediction, which could involve integrating game-theoretic approaches to anticipate the behavior of nearby human-driven vehicles. Moreover, integrating a visible risk assessment module could assist in improving lane-change decisions during merging.

Roundabouts present additional challenges, involving complex decisions in a continuous circular movement. Unlike linear movements on highways, roundabouts demand coordination with multiple vehicles simultaneously entering or exiting at different points. Effective decision-making in these environments could benefit from integrating dynamic priority systems that allow the AV to assess the intentions and priority of other vehicles, thereby navigating the roundabout efficiently. The incorporation of a reinforcement learning policy specifically trained on managing entry and exit at roundabouts would further improve navigation.

Unsignalized intersections introduce yet another layer of complexity due to the requirement for adaptive right-of-way decisions, pedestrian interaction, and simultaneous engagement with multiple agents. The introduction of priority assignment and right-of-way prediction mechanisms would enable more context-sensitive actions by AVs. Additionally, unsignalized intersections involve higher uncertainty due to non-standardized behaviors of other drivers. Thus, implementing uncertainty-aware learning mechanisms that focus on probabilistic modeling of interactions could significantly enhance performance in these situations.

Expanding the DRL-based model to include these varied scenarios would involve adopting hierarchical reinforcement learning, where high-level policies determine appropriate behaviors for each type of scenario (e.g., merging, entering/exiting roundabouts, stopping at intersections), and lower-level policies generate context-specific maneuvers.

5.2.3 Limited Racing Competitors

Currently, the thesis's focus on single-vehicle autonomous racing restricts the scope of the proposed PPO-C algorithm to solo driving tasks. Expanding PPO-C to include multi-vehicle competition would add complexity by introducing scenarios that require the vehicle to make strategic decisions in response to other competitors. Multi-vehicle racing presents a unique set of challenges that include overtaking, blocking, and cooperative racing strategies, all of which are crucial for achieving optimal performance in competitive settings. From a theoretical perspective, multi-agent interactions require algorithms to account for the potential behaviors and decisions of other vehicles, effectively simulating a game-theoretic environment. This involves modeling not only the physical aspects of driving but also the intentions and goals of other agents, introducing an additional layer of strategy to the decision-making process. In real-world applications, autonomous vehicles often need to navigate in environments with other vehicles, making split-second decisions in response to the actions of human drivers or other autonomous systems. Expanding PPO-C to a multi-agent framework would enhance its robustness, as it would learn to handle these complex interactions, improving decision-making under competitive pressure

and making the algorithm more applicable to real-world scenarios such as autonomous racing, cooperative transport systems, and emergency response. Ultimately, a multi-agent extension of PPO-C would deepen its strategic capabilities, allowing for a wider range of applications and increased resilience in competitive, high-stakes environments.

5.3 Future Work

5.3.1 Transition to 3D Simulation Environments

A key direction for future work involves transitioning from the 2D simulator used in this study to a high-fidelity 3D simulation environment, such as CARLA. Incorporating a 3D simulator would allow future research to benefit from more realistic vehicle dynamics, varied traffic scenarios, and complex environmental conditions, all of which are essential for closely simulating real-world complexities. This transition, however, presents significant computational challenges due to the increased data processing requirements associated with the richer sensory inputs and complex visual data in 3D environments. Additionally, as the simulation becomes more detailed, the model will need to address more sophisticated and nuanced road conditions, which may require the design and implementation of enhanced safety mechanisms to ensure the autonomous vehicle operates within safe bounds. The introduction of these safety constraints and the increased computational demands make this transition a longer-term objective, as it will involve extensive algorithmic adaptation, testing, and verification in a more demanding 3D environment.

5.3.2 DRL-Based Algorithms for Diverse Driving Scenarios

To extend the versatility of DRL-based algorithms, future work should focus on adapting and testing these algorithms in a wider variety of driving scenarios, such as on-ramp merging, roundabouts, intersections, and urban streets. Each of these scenarios presents unique challenges. For example, on-ramp merging requires coordination with other vehicles on the highway, while roundabouts necessitate continuous motion and multi-agent interaction in constrained spaces. Similarly, intersections involve stop-and-go decisions, right-of-way

considerations, and variable traffic signal timing. Addressing these diverse scenarios requires the development of scenario-specific reward structures, exploration strategies, and policy adjustments, allowing DRL-based algorithms to effectively handle each context. This line of research would enable autonomous driving systems to learn and adapt across a broader spectrum of situations, ultimately supporting the deployment of autonomous vehicles that can seamlessly operate in mixed and dynamic traffic environments.

5.3.3 Multi-Agent Competitive Models

A promising future direction is the extension of the PPO-C algorithm to multi-agent competitive settings. By expanding from a single-vehicle model to a multi-vehicle framework, the algorithm can be adapted to simulate competitive racing environments where autonomous vehicles must account for the actions of other agents in real-time. This would introduce an additional layer of strategic decision-making, where the vehicle would need to manage overtaking, blocking, and defensive driving maneuvers in response to other competitors. Implementing PPO-C in a multi-agent framework would involve leveraging game-theoretic principles to simulate interactions between autonomous vehicles, allowing for more sophisticated tactics and coordination. This approach could not only enhance the robustness of autonomous racing algorithms but also provide valuable insights for real-world applications, such as autonomous convoy driving, collaborative transportation, and high-stakes rescue missions. Multi-agent models would also foster innovation in competitive autonomous vehicle applications, supporting the development of algorithms capable of adapting to real-time interactions and pressures within a dynamic, multi-vehicle context.

5.3.4 Failure Cases

While the proposed algorithms demonstrated significant improvements in autonomous driving decision-making, some limitations were observed during evaluation. One notable failure case involved the RBDQN-CS algorithm, which exhibited increased collision rates when the speed limitation exceeded 25 m/s in highway driving scenarios. This issue arises from the risk-attention mechanism's inability to adapt quickly to dynamic traffic changes at higher speeds. To address this, future work could integrate adaptive risk perception

based on real-time traffic flow and implement speed-dependent safety margins to balance efficiency and safety under varying speed conditions. Similarly, the PPO-C algorithm encountered low-speed driving in multi-curve scenarios, particularly when consecutive sharp turns appeared within a short distance. This behavior was linked to the over-conservative reward structure, where safety was prioritized over speed in complex environments. To overcome this limitation, future improvements could involve dynamic reward shaping, encouraging the agent to maintain efficient speeds while navigating complex track sections safely. Additionally, integrating path prediction modules could help the agent anticipate upcoming curves and optimize its speed accordingly.

5.3.5 Implementation of Developed Models in Real-World Autonomous Driving

Given that real-world testing is not included in this thesis, this subsection summarizes the progress of developed models in real-world autonomous driving to support future research efforts. Deploying DRL models for real-world autonomous driving involves multiple critical considerations. One major challenge is the transferability of policies trained in simulators, such as CARLA [248] and AirSim [249], to real-world environments—a challenge commonly referred to as the sim-to-real gap. In [250], a real-world-like simulator called DriveDreamer is introduced, while a systematic approach for physical-world testing of autonomous driving systems, known as DeepBillboard, is proposed in [251]. To improve AVs' responses to challenging corner cases, a dataset capturing real-world-like corner cases was created in [252]. Techniques such as domain randomization and domain adaptation have also been proposed to address these challenges [253–255]. Additionally, due to constraints in the real-world testing environment, some studies have employed real-world data for training or evaluating the algorithms [256–259]. Furthermore, weather-related effects are considered in [260], thereby improving the vehicles' responses under abnormal weather conditions.

Safety and reliability remain paramount, necessitating rigorous testing of DRL-based models against unpredictable edge cases and interactions with human-driven vehicles. Recent approaches—including scenario-based testing and adversarial training—aim to ensure robust performance under diverse driving conditions [261–263]. Additionally, modular DRL architectures that integrate traditional control mechanisms have further enhanced safety by providing structured responses to high-risk scenarios [264].

Computational constraints represent another key issue, as real-time decision-making demands efficient models suitable for onboard deployment. Techniques such as network pruning and structured model compression have demonstrated significant promise in reducing model complexity without compromising performance [265, 266].

Finally, regulatory compliance and ethical considerations are critical for practical deployment. DRL-based decision-making frameworks must align with existing traffic regulations, ethical norms, and liability considerations. Current discussions and guidelines emphasize the need for clearer frameworks for ethical decision-making, accountability, and transparency in autonomous driving policies [267–269]. Addressing these real-world implementation considerations and incorporating the proposed DDTUI-based framework are essential steps for successfully transitioning DRL models from simulated environments to safe, efficient, and socially acceptable autonomous driving systems in the real world.

Bibliography

[1] Department for Transport, "Road accidents and safety statistics," https://www.gov.uk/government/collections/road-accidents-and-safety-statistics, 2023, accessed: 2024-04-28.

[2] D. Omeiza, H. Webb, M. Jirotka, and L. Kunze, "Explanations in autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 142–10 162, 2021.

[3] H. A. Ignatious, M. Khan *et al.*, "An overview of sensors in autonomous vehicles," *Procedia Computer Science*, vol. 198, pp. 736–741, 2022.

[4] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz *et al.*, "Self-driving cars: A survey," *Expert systems with applications*, vol. 165, p. 113816, 2021.

[5] J. Pérez, V. Milanés *et al.*, "Autonomous driving manoeuvres in urban road traffic environment: a study on roundabouts," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 13 795–13 800, 2011.

[6] Z. Lin, Q. Zhang, Z. Tian, P. Yu, and J. Lan, "Dpl-slam: Enhancing dynamic point-line slam through dense semantic methods," *IEEE Sensors Journal*, 2024.

[7] Z. Lin, Q. Zhang, Z. Tian, P. Yu, Z. Ye, H. Zhuang, and J. Lan, "Slam2: Simultaneous localization and multimode mapping for indoor dynamic environments," *Pattern Recognition*, p. 111054, 2024.

[8] World Health Organization, "Road traffic injuries," https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, 2023, accessed: 2024-04-28.

[9] H. Vijayakumar, D. Zhao *et al.*, "A holistic safe planner for automated driving considering interaction with human drivers," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 2061–2076, 2023.

[10] J. Löfberg, "Minimax approaches to robust model predictive control," vol. 812, pp. 29–37, 2003.

[11] S. V. Raković, "Model predictive control: classical, robust, and stochastic [bookshelf]," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 102–105, 2016.

[12] P. Hang, C. Lv, C. Huang, J. Cai, Z. Hu, and Y. Xing, "An integrated framework of decision making and motion planning for autonomous vehicles considering social behaviors," *IEEE transactions on vehicular technology*, vol. 69, no. 12, pp. 14 458–14 469, 2020.

[13] Z. Lin, Z. Tian *et al.*, "Enhanced visual slam for collision-free driving with lightweight autonomous cars," *Sensors*, vol. 24, no. 19, p. 6258, 2024.

[14] P. Bhattacharya and M. L. Gavrilova, "Voronoi diagram in optimal path planning," *International Symposium on Voronoi Diagrams in Science and Engineering*, pp. 38–47, 2007.

[15] M. Abdel-Aty and S. Ding, "A matched case-control analysis of autonomous vs human-driven vehicle accidents," *Nature Communications*, vol. 15, no. 1, p. 4931, 2024.

[16] Q. Yao, Z. Zheng *et al.*, "Path planning method with improved artificial potential field—a reinforcement learning perspective," *IEEE access*, vol. 8, pp. 135 513–135 523, 2020.

[17] H. H. Triharminto, O. Wahyunggoro *et al.*, "A novel of repulsive function on artificial potential field for robot path planning," *International Journal of Electrical and Computer Engineering*, vol. 6, no. 6, p. 3262, 2016.

[18] H. H. Triharminto, O. Wahyunggoro, T. B. Adji, A. Cahyadi, I. Ardiyanto, and Iswanto, "Local information using stereo camera in artificial potential field based path planning," *IAENG International Journal of Computer Science*, vol. 44, no. 3, pp. 316–326, 2017.

[19] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, no. 1, pp. 381–386, 2020.

[20] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[21] I. Muhammad and Z. Yan, "Supervised machine learning approaches: A survey." *ICTACT Journal on Soft Computing*, vol. 5, no. 3, 2015.

[22] M. Castelli, L. Vanneschi *et al.*, "Supervised learning: classification," *por Ranganathan, S., M. Grisbskov, K. Nakai y C. Schönbach*, vol. 1, pp. 342–349, 2018.

[23] R. Tian, S. Li *et al.*, "Adaptive game-theoretic decision making for autonomous vehicle control at roundabouts," *IEEE conference on decision and control*, pp. 321–326, 2018.

[24] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[25] J. Lu, L. Han *et al.*, "Event-triggered deep reinforcement learning using parallel control: A case study in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 4, pp. 2821–2831, 2023.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[27] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," vol. 196, p. 800, 2016.

[28] K. Yeom, "Deep reinforcement learning based autonomous driving with collision free for mobile robots," *International Journal of Mechanical Engineering and Robotics Research*, vol. 11, no. 5, pp. 338–344, 2022.

[29] A. J. M. Muzahid, S. F. Kamarulzaman *et al.*, "Deep reinforcement learning-based driving strategy for avoidance of chain collisions and its safety efficiency analysis in autonomous vehicles," *IEEE Access*, vol. 10, pp. 43 303–43 319, 2022.

[30] C. Xu, W. Zhao *et al.*, "A Nash Q-learning based motion decision algorithm with considering interaction to traffic participants," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 621–12 634, 2020.

[31] K. Min, H. Kim *et al.*, "Deep Q learning based high level driving policy determination," *IEEE Intelligent Vehicles Symposium*, pp. 226–231, 2018.

[32] S. Gu, T. Lillicrap *et al.*, "Continuous deep Q-learning with model-based acceleration," in *Proceedings of the International Conference on Machine Learning*, 2016, pp. 2829–2838.

[33] H. Wei, X. Liu *et al.*, "Mixed-autonomy traffic control with proximal policy optimization," in *Proceedings of the IEEE Vehicular Networking Conference*, 2019, pp. 1–8.

[34] F. Ye, X. Cheng *et al.*, "Automated lane change strategy using proximal policy optimization-based deep reinforcement learning," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2020, pp. 1746–1752.

[35] G. Dulac-Arnold, R. Evans *et al.*, "Deep reinforcement learning in large discrete action spaces. arxiv 2015," *arXiv preprint arXiv:1512.07679*.

[36] Z. Tian, D. Zhao *et al.*, "Efficient and balanced exploration-driven decision making for autonomous racing using local information," *IEEE Transactions on Intelligent Vehicles*, 2024.

[37] Z. Tian, D. Zhao, Z. Lin, D. Flynn, W. Zhao, and D. Tian, "Balanced reward-inspired reinforcement learning for autonomous vehicle racing," in *6th Annual Learning for Dynamics & Control Conference*. PMLR, 2024, pp. 628–640.

[38] G. Basile, A. Petrillo, and S. Santini, "DDPG based end-to-end driving enhanced with safe anomaly detection functionality for autonomous vehicles," in *Proceedings of the IEEE International Conference on Metrology for Extended Reality, Artificial Intelligence and Neural Engineering*, 2022, pp. 248–253.

[39] M. A. Hebaish, A. Hussein *et al.*, "Towards safe and efficient modular path planning using twin delayed DDPG," in *Proceedings of the IEEE Vehicular Technology Conference*, 2022, pp. 1–7.

[40] H. Gao, Y. Li, K. Long, M. Yang, and Y. Shen, "A survey for foundation models in autonomous driving," *arXiv preprint arXiv:2402.01105*, 2024, provides an extensive survey of foundation models in autonomous driving, discussing scalability, integration challenges, and potential applications.

[41] J. Mao, J. Ye, Y. Qian, M. Pavone, and Y. Wang, "A language agent for autonomous driving," *arXiv preprint arXiv:2311.10813*, 2023, proposes an architecture where a large language model is integrated as a cognitive agent for high-level planning and decision-making in autonomous driving.

[42] H. Sha, Y. Mu, Y. Jiang, L. Chen, C. Xu, P. Luo, S. E. Li, M. Tomizuka, W. Zhan, and M. Ding, "LanguageMPC: Large language models as decision makers for autonomous driving," *arXiv preprint arXiv:2310.03026*, 2023, leverages a large language model for decision-making in complex driving scenarios, formulated as a model-predictive control problem for enhanced safety and reasoning.

[43] J.-J. Hwang, R. Xu, H. Lin, W.-C. Hung, J. Ji, K. Choi, D. Huang, T. He, P. Covington, B. Sapp, J. Guo, D. Anguelov, and M. Tan, "EMMA: End-to-end multimodal model for autonomous driving," *arXiv preprint arXiv:2410.23262*, 2024, introduces a multimodal foundation model integrating vision and language to jointly address perception and planning tasks in autonomous driving systems.

[44] X. Wang, Z. Zhu, G. Huang, X. Chen, J. Zhu, and J. Lu, "Drivedreamer: Towards real-world-driven world models for autonomous driving," *arXiv preprint arXiv:2309.09777*, 2023, presents a diffusion-based generative world model trained on real driving logs to simulate realistic driving scenarios and address the sim-to-real gap.

[45] L. Abualigah, S. Ekinci *et al.*, "Modified elite opposition-based artificial hummingbird algorithm for designing fopid controlled cruise control system." *Intelligent Automation & Soft Computing*, vol. 38, no. 2, 2023.

[46] S. Tang, Z. Zhang, Y. Zhang, J. Zhou, Y. Guo, S. Liu, S. Guo, Y.-F. Li, L. Ma, Y. Xue *et al.*, "A survey on automated driving system testing: Landscapes and trends," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 5, pp. 1–62, 2023.

[47] K. Yang, X. Tang *et al.*, "Towards robust decision-making for autonomous driving on highway," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 9, pp. 11 251–11 263, 2023.

[48] J. Wu, Z. Song *et al.*, "Deep reinforcement learning-based energy-efficient decision-making for autonomous electric vehicle in dynamic traffic environments," *IEEE Transactions on Transportation Electrification*, vol. 10, no. 1, pp. 875–887, 2023.

[49] Y. Fu, C. Li *et al.*, "An incentive mechanism of incorporating supervision game for federated learning in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 14 800–14 812, 2023.

[50] W. Yue, X. Wu, C. Li, N. Cheng, P. Duan, and Z. Han, "Navigating the impact of connected and automated vehicles on mixed traffic efficiency: A driving behavior perspective," *IEEE Internet of Things Journal*, 2024.

[51] B. Toghi, R. Valiente, D. Sadigh, R. Pedarsani, and Y. P. Fallah, "Social coordination and altruism in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 791–24 804, 2022.

[52] Y. Liu, X. Zhao, Y. Tian, and J. Sun, "Sociality probe: Game-theoretic inverse reinforcement learning for modeling and quantifying social patterns in driving interaction," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[53] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi, "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability," *Computer Science Review*, vol. 37, p. 100270, 2020.

[54] F.-L. Fan, J. Xiong, M. Li, and G. Wang, "On interpretability of artificial neural networks: A survey," *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 5, no. 6, pp. 741–760, 2021.

[55] C. D. M. Papageorgiou and T. McLean, "Integrated traffic-responsive urban corridor control strategy in glasgow, scotland," *Transportation Research Record*, vol. 1, p. 727.

[56] D. Karas, "Highway to inequity: the disparate impact of the interstate highway system on poor and minority communities in american cities," *New Visions for Public Affairs*, vol. 7, no. April, pp. 9–21, 2015.

[57] M. Gross, "Speed tourism: The german autobahn as a tourist destination and location of "unruly rules"," *Tourist Studies*, vol. 20, no. 3, pp. 298–313, 2020.

[58] J. P. Leisch, "Freeway and interchange design: A historical perspective," *Transportation Research Record*, pp. 60–60, 1993.

[59] G. Davis, M. Contreras-Sweet *et al.*, "Ramp meter design manual," *Traffic Operation Program, Department of California Highway Patrol*, 2000.

[60] A. Pratelli and R. R. Souleyrette, "Visibility, perception and roundabout safety," *WIT Transactions on the built environment*, vol. 107, pp. 577–588, 2009.

[61] M. Naderi, M. Papageorgiou *et al.*, "Automated vehicle driving on large lane-free roundabouts," in *IEEE 25th International Conference on Intelligent Transportation Systems*, 2022, pp. 1528–1535.

[62] J. G. Bared, P. K. Edara *et al.*, "Design and operational performance of double crossover intersection and diverging diamond interchange," *Transportation Research Record*, vol. 1912, no. 1, pp. 31–38, 2005.

[63] J. York and T. Maze, "Economic evaluation of truck collision warning systems," *Transportation Research Circular*, vol. 475, pp. 46–50, 1997.

[64] I. C. Burnett, *Traffic Collisions in North Carolina: Weather, Human Factors, and Economic Analysis, 2013 to 2019*. North Carolina State University, 2023.

[65] J. Wang, J. Wu, X. Zheng, D. Ni, and K. Li, "Driving safety field theory modeling and its application in pre-collision warning system," *Transportation research part C: emerging technologies*, vol. 72, pp. 306–324, 2016.

[66] R. Nahata, D. Omeiza, R. Howard, and L. Kunze, "Assessing and explaining collision risk in dynamic environments for autonomous driving safety," in *2021 IEEE international intelligent transportation systems conference (ITSC)*. IEEE, 2021, pp. 223–230.

[67] M. Wang, L. Zhang, Z. Zhang, and Z. Wang, "A hybrid trajectory planning strategy for intelligent vehicles in on-road dynamic scenarios," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 2832–2847, 2023.

[68] A. Botros and S. L. Smith, "Spatio-temporal lattice planning using optimal motion primitives," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[69] W. Chen, Y. Chen *et al.*, "Motion planning using feasible and smooth tree for autonomous driving," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 5, pp. 6270–6282, 2024.

[70] F. Bouchard, S. Sedwards, and K. Czarnecki, "A rule-based behaviour planner for autonomous driving," in *Proceeding of the International Joint Conference on Rules and Reasoning*. Springer, 2022, pp. 263–279.

[71] R. Gajjar and D. Mohandas, "Critical assessment of road capacities on urban roads–a mumbai case-study," *Transportation Research Procedia*, vol. 17, pp. 685–692, 2016.

[72] M. A. S. Kamal, T. Hayakawa, and J.-i. Imura, "Road-speed profile for enhanced perception of traffic conditions in a partially connected vehicle environment," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 6824–6837, 2018.

[73] M. A. S. Kamal, S. Taguchi, and T. Yoshimura, "Efficient driving on multilane roads under a connected vehicle environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2541–2551, 2016.

[74] D. A. Hensher, "Valuation of travel time savings," in *A handbook of transport economics*. Edward Elgar Publishing, 2011.

[75] F. Steck, V. Kolarova, F. Bahamonde-Birke, S. Trommer, and B. Lenz, "How autonomous driving may affect the value of travel time savings for commuting," *Transportation research record*, vol. 2672, no. 46, pp. 11–20, 2018.

[76] C. Zhai, F. Luo, Y. Liu, and Z. Chen, "Ecological cooperative look-ahead control for automated vehicles travelling on freeways with varying slopes," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1208–1221, 2018.

[77] S. A. Birrell, M. Fowkes *et al.*, "Effect of using an in-vehicle smart driving aid on real-world driver performance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1801–1810, 2014.

[78] C. Sun, J. Guanetti *et al.*, "Optimal eco-driving control of connected and autonomous vehicles through signalized intersections," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3759–3773, 2020.

[79] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 519–16 529.

[80] R. Xu, J. Joshi *et al.*, "Nn-emd: Efficiently training neural networks using encrypted multi-sourced datasets," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2807–2820, 2021.

[81] H. Touvron, P. Bojanowski *et al.*, "Resmlp: Feedforward networks for image classification with data-efficient training," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 4, pp. 5314–5321, 2022.

[82] C. M. Martinez, M. Heucke, F.-Y. Wang, B. Gao, and D. Cao, "Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 666–676, 2017.

[83] X. Li, W. Wang, and M. Roetting, "Estimating driver's lane-change intent considering driving style and contextual traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3258–3271, 2018.

[84] D. Xu, H. Zhao, F. Guillemard, S. Geronimi, and F. Aioun, "Aware of scene vehicles—probabilistic modeling of car-following behaviors in real-world traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2136–2148, 2018.

[85] P. Jardin, I. Moisidis, S. S. Zetina, and S. Rinderknecht, "Rule-based driving style classification using acceleration data profiles," in *Proc. IEEE ITSC*, 2020, pp. 1–6.

[86] R. Vogel, F. Schmidsberger, A. Kühn, K. A. Schneider *et al.*, "You can't drive my car-a method to fingerprint individual driving styles in a sim-racing setting," in *Proc. ICECET*, 2022, pp. 1–9.

[87] F. Lateef, M. Kas *et al.*, "Saliency heat-map as visual attention for autonomous driving using generative adversarial network (GAN)," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5360–5373, 2022.

[88] N. Ding, C. Zhang *et al.*, "Saliendet: A saliency-based feature enhancement algorithm for object detection for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 2624–2635, 2023.

[89] Z. Cui, M. Li *et al.*, "An interpretation framework for autonomous vehicles decision-making via shap and rf," in *Proceeding of the CAA International Conference on Vehicular Control and Intelligence*, 2022, pp. 1–7.

[90] B. Gyevnar, C. Wang *et al.*, "Causal explanations for sequential decision-making in multi-agent systems," *arXiv preprint arXiv:2302.10809*, 2023.

[91] P. M. Dassanayake, A. Anjum *et al.*, "A deep learning based explainable control system for reconfigurable networks of edge devices," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 7–19, 2021.

[92] M. Zemni, M. Chen *et al.*, "Octet: Object-aware counterfactual explanations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 062–15 071.

[93] Z. Chen, F. Xiao, F. Guo, and J. Yan, "Interpretable machine learning for building energy management: A state-of-the-art review," *Advances in Applied Energy*, vol. 9, p. 100123, 2023.

[94] D. Leslie, "Understanding artificial intelligence ethics and safety," *arXiv preprint arXiv:1906.05684*, 2019.

[95] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, "A benchmark for interpretability methods in deep neural networks," *Advances in neural information processing systems*, vol. 32, 2019.

[96] R. Tomsett, D. Harborne, S. Chakraborty, P. Gurram, and A. Preece, "Sanity checks for saliency metrics," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 6021–6029.

[97] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *Advances in neural information processing systems*, vol. 31, 2018.

[98] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3681–3688.

[99] A. A. Ismail, M. Gunady, L. Pessoa, H. Corrada Bravo, and S. Feizi, "Input-cell attention reduces vanishing saliency of recurrent neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[100] M. Wu, M. Hughes, S. Parbhoo, M. Zazzi, V. Roth, and F. Doshi-Velez, "Beyond sparsity: Tree regularization of deep models for interpretability," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[101] T. Speith and M. Langer, "A new perspective on evaluation methods for explainable artificial intelligence (xai)," in *IEEE 31st International Requirements Engineering Conference Workshops*. IEEE, 2023, pp. 325–331.

[102] A. Baheri, S. Nageshrao *et al.*, "Deep reinforcement learning with enhanced safety for autonomous highway driving," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1550–1555.

[103] M. Kaushik, V. Prasad *et al.*, "Overtaking maneuvers in simulated highway driving using deep reinforcement learning," in *2018 IEEE intelligent vehicles symposium (iv)*. IEEE, 2018, pp. 1885–1890.

[104] N. Albarella, D. G. Lui *et al.*, "A hybrid deep reinforcement learning and optimal control architecture for autonomous highway driving," *Energies*, vol. 16, no. 8, p. 3490, 2023.

[105] H. Meng, H. Bin *et al.*, "Optimizing distributed energy system with an enhanced reinforcement learning–model predictive control algorithm," *Available at SSRN 4876862*.

[106] Z. Lin, Z. Tian *et al.*, "A conflicts-free, speed-lossless kan-based reinforcement learning decision system for interactive driving in roundabouts," *arXiv preprint arXiv:2408.08242*, 2024.

[107] J. Gómez-Romero, "Explaining deep reinforcement learning-based methods for control of building hvac systems," *Methods*, vol. 15, p. 52.

[108] S. Shalev-Shwartz, S. Shammah *et al.*, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[109] C. Yu, X. Wang, J. Hao, and Z. Feng, "Reinforcement learning for cooperative overtaking," in *Proceedings of the 18th international conference on autonomous agents and multiagent systems*, 2019, pp. 341–349.

[110] M. B. Ozcelik, B. Agin *et al.*, "Decision making for autonomous driving in a virtual highway environment based on generative adversarial imitation learning," in *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2023, pp. 1–6.

[111] S. Zhang, Y. Wu *et al.*, "Spatial attention for autonomous decision-making in highway scene," in *Proceeding of the Annual Conference of the Society of Instrument and Control Engineers of Japan*, 2020, pp. 1435–1440.

[112] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[113] S. Nageshrao, H. E. Tseng, and D. Filev, "Autonomous highway driving using deep reinforcement learning," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2019, pp. 2326–2331.

[114] J. Zhao, T. Qu *et al.*, "A deep reinforcement learning approach for autonomous highway driving," *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 542–546, 2020.

[115] W. Zhao, S. Gong, D. Zhao, F. Liu, N. Sze, M. Quddus, and H. Huang, "A spatial-state-based omni-directional collision warning system for intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[116] B. M. Albaba and Y. Yildiz, "Driver modeling through deep reinforcement learning and behavioral game theory," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 2, pp. 885–892, 2021.

[117] J. Yang, A. Nakhaei *et al.*, "Cm3: Cooperative multi-goal multi-stage multi-agent reinforcement learning," *arXiv preprint arXiv:1809.05188*, 2018.

[118] M. Schutera, N. Goby *et al.*, "Transfer learning versus multi-agent learning regarding distributed decision-making in highway traffic," *arXiv preprint arXiv:1810.08515*, 2018.

[119] X. Xu, L. Zuo *et al.*, "A reinforcement learning approach to autonomous decision making of intelligent vehicles on highways," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 10, pp. 3884–3897, 2018.

[120] Z. Bai, W. Shangguan *et al.*, "Deep reinforcement learning based high-level driving behavior decision-making model in heterogeneous traffic," in *2019 Chinese Control Conference (CCC)*. IEEE, 2019, pp. 8600–8605.

[121] T. Liu, Q. Liu *et al.*, "Combining deep reinforcement learning with rule-based constraints for safe highway driving," in *Proceeding of the China Automation Congress*, 2022, pp. 2785–2790.

[122] J. Liao, T. Liu *et al.*, "Decision-making strategy on highway for autonomous vehicles using deep reinforcement learning," *IEEE Access*, vol. 8, pp. 177 804–177 814, 2020.

[123] W. Yuan, M. Yang *et al.*, "Multi-reward architecture based reinforcement learning for highway driving policies," in *Proceeding of the IEEE Intelligent Transportation Systems Conference*, 2019, pp. 3810–3815.

[124] S. Aradi, T. Becsi *et al.*, "Policy gradient based reinforcement learning approach for autonomous highway driving," in *Proceeding of the IEEE Conference on Control Technology and Applications*. IEEE, 2018, pp. 670–675.

[125] H. Wang, S. Yuan *et al.*, "Tactical driving decisions of unmanned ground vehicles in complex highway environments: A deep reinforcement learning approach," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 4, pp. 1113–1127, 2021.

[126] A. M. Naveen, R. Ravish *et al.*, "Distributional reinforcement learning for automated driving vehicle," in *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*, 2022, pp. 1–6.

[127] J. Wang, T. Yang *et al.*, "Learning an efficient and safe policy for highway driving using supervised learning and reinforcement learning," in *Proceeding of the International Conference on Real-time Computing and Robotics (RCAR)*, 2019, pp. 112–117.

[128] K. Lv, X. Pei, C. Chen, and J. Xu, "A safe and efficient lane change decision-making strategy of autonomous driving based on deep reinforcement learning," *Mathematics*, vol. 10, no. 9, p. 1551, 2022.

[129] R. Rădulescu, M. Legrand *et al.*, "Deep multi-agent reinforcement learning in a homogeneous open population," in *Artificial Intelligence: 30th Benelux Conference, BNAIC 2018,'s-Hertogenbosch, The Netherlands, November 8–9, 2018, Revised Selected Papers 30*. Springer, 2019, pp. 90–105.

[130] C. Yu, X. Wang *et al.*, "Distributed multiagent coordinated learning for autonomous driving in highways based on dynamic coordination graphs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 735–748, 2020.

[131] M. Kaushik, N. Singhania *et al.*, "Parameter sharing reinforcement learning architecture for multi agent driving," in *Proceedings of the 2019 4th International Conference on Advances in Robotics*, 2019, pp. 1–7.

[132] M. Molaie, A. Amirkhani *et al.*, "Auto-driving policies in highway based on distributional deep reinforcement learning," in *Proceeding of the International Conference on Pattern Recognition and Image Analysis*, 2021, pp. 1–6.

[133] G. Chen, Y. Zhang *et al.*, "Attention-based highway safety planner for autonomous driving via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 1, pp. 162–175, 2024.

[134] K. Min, H. Kim *et al.*, "Deep distributional reinforcement learning based high-level driving policy determination," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 3, pp. 416–424, 2019.

[135] B. Gangopadhyay, H. Soora *et al.*, "Hierarchical program-triggered reinforcement learning agents for automated driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 902–10 911, 2022.

[136] S. Cheng, B. Yang, Z. Wang, and K. Nakano, "Spatio-temporal image representation and deep-learning-based decision framework for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 866–24 875, 2022.

[137] B. Liu, Y. Tang *et al.*, "A deep reinforcement learning approach for ramp metering based on traffic video data," *Journal of Advanced Transportation*, vol. 2021, no. 1, p. 6669028, 2021.

[138] M. Yang, Z. Li *et al.*, "A deep reinforcement learning-based ramp metering control framework for improving traffic operation at freeway weaving sections," in *Proceedings of the Transportation Research Board 98th Annual Meeting, Washington, DC, USA*, 2019, pp. 13–17.

[139] P. Wang and C.-Y. Chan, "Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge," in *Proceedings of the International Conference on Intelligent Transportation Systems*, 2017, pp. 1–6.

[140] Y. Lin, J. McPhee *et al.*, "Anti-jerk on-ramp merging using deep reinforcement learning," in *Proceedings of the Intelligent Vehicles Symposium*. IEEE, 2020, pp. 7–14.

[141] F. Deng, J. Jin *et al.*, "Advanced self-improving ramp metering algorithm based on multi-agent deep reinforcement learning," in *Proceeding of the Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3804–3809.

[142] M. Cheng, C. Zhang *et al.*, "Adaptive coordinated variable speed limit between highway mainline and on-ramp with deep reinforcement learning," *Journal of Advanced Transportation*, vol. 2022, no. 1, p. 2435643, 2022.

[143] Z. Hu and W. Ma, "guided deep reinforcement learning for coordinated ramp metering and perimeter control in large scale networks," *Transportation research part C: emerging technologies*, vol. 159, p. 104461, 2024.

[144] D. Deng, B. Yu *et al.*, "Automated traffic state optimization in the weaving area of urban expressways by a reinforcement learning-based cooperative method of channelization and ramp metering," *Journal of Advanced Transportation*, vol. 2023, no. 1, p. 4771946, 2023.

[145] S. Zhou, W. Zhuang *et al.*, "Cooperative on-ramp merging control of connected and automated vehicles: Distributed multi-agent deep reinforcement learning approach," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 402–408.

[146] C. Wang, Y. Xu *et al.*, "Integrated traffic control for freeway recurrent bottleneck based on deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 522–15 535, 2022.

[147] X. Qi, L. Zhang *et al.*, "Learning-based mpc for autonomous motion planning at freeway off-ramp diverging," *IEEE Transactions on Intelligent Vehicles*, pp. 1–11, 2024.

[148] Z. e. a. Kherroubi, S. Aknine, and R. Bacha, "Novel decision-making strategy for connected and autonomous vehicles in highway on-ramp merging," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12 490–12 502, 2022.

[149] X. Zhang, L. Wu *et al.*, "High-speed ramp merging behavior decision for autonomous vehicles based on multiagent reinforcement learning," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 22 664–22 672, 2023.

[150] D. Chen, M. R. Hajidavalloo *et al.*, "Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 11, pp. 11 623–11 638, 2023.

[151] X. He, B. Lou *et al.*, "Robust decision making for autonomous vehicles at highway on-ramps: A constrained adversarial reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 4103–4113, 2023.

[152] M. Li, Z. Li *et al.*, "Enhancing cooperation of vehicle merging control in heavy traffic using communication-based soft actor-critic algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 6491–6506, 2023.

[153] J. Chen, B. Yuan *et al.*, "Model-free deep reinforcement learning for urban autonomous driving," in *Proceeding of the intelligent transportation systems conference.* IEEE, 2019, pp. 2765–2771.

[154] G. Bacchiani, D. Molinari, and M. Patander, "Microscopic traffic simulation by cooperative multi-agent deep reinforcement learning," *arXiv preprint arXiv:1903.01365*, 2019.

[155] B. Montgomery, C. Muise *et al.*, "Hierarchical deep reinforcement learning with cross-attention and planning for autonomous roundabout navigation," in *Proceeding of the Canadian Conference on Electrical and Computer Engineering*, 2024, pp. 417–423.

[156] Y. Tian, M. Han, L. Zhang, W. Liu, J. Wang, and W. Pan, "Variational constrained reinforcement learning with application to planning at roundabout," 2020.

[157] W. Wang, L. Jiang *et al.*, "Imitation learning based decision-making for autonomous vehicle control at traffic roundabouts," *Multimedia Tools and Applications*, vol. 81, no. 28, pp. 39 873–39 889, 2022.

[158] L. Ferrarotti, M. Luca, G. Santin, G. Previati, G. Mastinu, M. Gobbi, E. Campi, L. Uccello, A. Albanese, P. Zalaya *et al.*, "Autonomous and human-driven vehicles interacting in a roundabout: A quantitative and qualitative evaluation," *IEEE Access*, 2024.

[159] Y. Zhang, B. Gao *et al.*, "Adaptive decision-making for automated vehicles under roundabout scenarios using optimization embedded reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5526–5538, 2021.

[160] F. Konstantinidis, M. Sackmann, O. De Candido, U. Hofmann, J. Thielecke, and W. Utschick, "Parameter sharing reinforcement learning for modeling multi-agent driving behavior in roundabout scenarios," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 1974–1981.

[161] R. Nakaya, T. Harada *et al.*, "Emergence of cooperative automated driving control at roundabouts using deep reinforcement learning," in *Proceeding of the Annual Conference of the Society of Instrument and Control Engineers*, 2023, pp. 97–102.

[162] H. Yuan, P. Li *et al.*, "Safe, efficient, comfort, and energy-saving automated driving through roundabout based on deep reinforcement learning," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC).* IEEE, 2023, pp. 6074–6079.

[163] W. Wang, F. Hui *et al.*, "Deep reinforcement learning method for trajectory planning of connected and autonomous vehicles in the roundabout lane-changing scenario," in *Proceeding of the International Symposium on Computer Technology and Information Science*, 2024, pp. 168–173.

[164] A. P. Capasso, G. Bacchiani *et al.*, "From simulation to real world maneuver execution using deep reinforcement learning," in *2020 IEEE Intelligent Vehicles Symposium (IV).* IEEE, 2020, pp. 1570–1575.

[165] ——, "Intelligent roundabout insertion using deep reinforcement learning," *arXiv preprint arXiv:2001.00786*, 2020.

[166] S. Alighanbari and N. L. Azad, "Deep reinforcement learning with nmpc assistance nash switching for urban autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 3, pp. 2604–2615, 2023.

[167] B. Peng, M. F. Keskin *et al.*, "Connected autonomous vehicles for improving mixed traffic efficiency in unsignalized intersections with deep reinforcement learning," *Communications in Transportation Research*, vol. 1, p. 100017, 2021.

[168] A. Pozzi, S. Bae, Y. Choi, F. Borrelli, D. M. Raimondo, and S. Moura, "Ecological velocity planning through signalized intersections: A deep reinforcement learning approach," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 245–252.

[169] K.-F. Chu, A. Y. S. Lam, and V. O. K. Li, "Traffic signal control using end-to-end off-policy deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7184–7195, 2022.

[170] D. Quang Tran and S.-H. Bae, "Proximal policy optimization through a deep reinforcement learning framework for multiple autonomous vehicles at a non-signalized intersection," *Applied Sciences*, vol. 10, no. 16, p. 5722, 2020.

[171] Z. Bai, P. Hao, W. Shangguan, B. Cai, and M. J. Barth, "Hybrid reinforcement learning-based eco-driving strategy for connected and automated vehicles at signalized intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 850–15 863, 2022.

[172] R. Bautista-Montesano, R. Galluzzi *et al.*, "Autonomous navigation at unsignalized intersections: A coupled reinforcement learning and model predictive control approach," *Transportation research part C: emerging technologies*, vol. 139, p. 103662, 2022.

[173] D. Li, F. Zhu, T. Chen, Y. D. Wong, C. Zhu, and J. Wu, "Coor-plt: A hierarchical control model for coordinating adaptive platoons of connected and autonomous vehicles at signal-free intersections based on deep reinforcement learning," *Transportation Research Part C: Emerging Technologies*, vol. 146, p. 103933, 2023.

[174] S. Kai, B. Wang *et al.*, "A multi-task reinforcement learning approach for navigating unsignalized intersections," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1583–1588.

[175] B. Zhou, Q. Zhou, S. Hu, D. Ma, S. Jin, and D.-H. Lee, "Cooperative traffic signal control using a distributed agent-based deep reinforcement learning with incentive communication," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 8, pp. 10 147–10 160, 2024.

[176] W. X. Hu, H. Ishihara, C. Chen, A. Shalaby, and B. Abdulhai, "Deep reinforcement learning two-way transit signal priority algorithm for optimizing headway adherence and speed," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 7920–7931, 2023.

[177] A. Lombard, A. Noubli, A. Abbas-Turki, N. Gaud, and S. Galland, "Deep reinforcement learning approach for v2x managed intersections of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 7178–7189, 2023.

[178] D. Li, J. Wu *et al.*, "Adaptive traffic signal control model on intersections based on deep reinforcement learning," *Journal of Advanced Transportation*, vol. 2020, no. 1, p. 6505893, 2020.

[179] H. Shu, T. Liu, X. Mu, and D. Cao, "Driving tasks transfer using deep reinforcement learning for decision-making of autonomous vehicles in unsignalized intersection," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 41–52, 2021.

[180] H. Seong, C. Jung, S. Lee, and D. H. Shim, "Learning to drive at unsignalized intersections using attention-based deep reinforcement learning," in *Proceeding of the International Intelligent Transportation Systems Conference*, 2021, pp. 559–566.

[181] C.-J. Hoel, T. Tram, and J. Sjöberg, "Reinforcement learning with uncertainty estimation for tactical decision-making in intersections," in *2020 IEEE 23rd international conference on intelligent transportation systems (ITSC)*. IEEE, 2020, pp. 1–7.

[182] I. Osband, J. Aslanides, and A. Cassirer, "Randomized prior functions for deep reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[183] G.-P. Antonio and C. Maria-Dolores, "Multi-agent deep reinforcement learning to manage connected autonomous vehicles at tomorrow's intersections," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7033–7043, 2022.

[184] W. Xiao, Y. Yang, X. Mu, Y. Xie, X. Tang, D. Cao, and T. Liu, "Decision-making for autonomous vehicles in random task scenarios at unsignalized intersection using deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 6, pp. 7812–7825, 2024.

[185] Z. Guo, Y. Wu, L. Wang, and J. Zhang, "Coordination for connected and automated vehicles at non-signalized intersections: A value decomposition-based multiagent deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 3025–3034, 2023.

[186] J. Fang, F. Wang, J. Xue, and T.-S. Chua, "Behavioral intention prediction in driving scenes: A survey," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–22, 2024.

[187] W. Zhang, Y. Zhang *et al.*, "Danger-aware adaptive composition of drl agents for self-navigation," *Unmanned Systems*, vol. 9, no. 01, pp. 1–9, 2021.

[188] M. Roderick, J. MacGlashan, and S. Tellex, "Implementing the deep q-network," *arXiv preprint arXiv:1711.07478*, 2017.

[189] M. Schuurmans *et al.*, "Safe, learning-based mpc for highway driving under lane-change uncertainty: A distributionally robust approach," *Artificial Intelligence*, vol. 320, p. 103920, 2023.

[190] P. Wang, C.-Y. Chan *et al.*, "Automated driving maneuvers under interactive environment based on deep reinforcement learning," *arXiv preprint arXiv:1803.09200*, 2018.

[191] Y. Jaafra, J. L. Laurent *et al.*, "Robust reinforcement learning for autonomous driving," 2019.

[192] L. Lyu, Y. Shen *et al.*, "The advance of reinforcement learning and deep reinforcement learning," in *Proceedings of the IEEE International Conference on Electrical Engineering, Big Data and Algorithms*, 2022, pp. 644–648.

[193] K. Yang, X. Tang *et al.*, "Towards robust decision-making for autonomous driving on highway," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 9, pp. 11 251–11 263, 2023.

[194] B. Peng, Y. Xie *et al.*, "Communication scheduling by deep reinforcement learning for remote traffic state estimation with bayesian inference," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 4287–4300, 2022.

[195] S. Siboo, A. Bhattacharyya *et al.*, "An empirical study of DDPG and PPO-based reinforcement learning algorithms for autonomous driving," *IEEE Access*, vol. 11, pp. 125 094–125 108, 2023.

[196] L. Li, Y. Lv *et al.*, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, 2016.

[197] J. Wang and S. Li, "Self-attention mechanism based system for dcase2018 challenge task1 and task4," *Proc. DCASE Challenge*, pp. 1–5, 2018.

[198] Z. Liu, J. Chen *et al.*, "Methodology of hierarchical collision avoidance for high-speed self-driving vehicle based on motion-decoupled extraction of scenarios," *IET Intelligent Transport Systems*, vol. 14, no. 3, pp. 172–181, 2020.

[199] M. Pourabdollah, E. Bjärkvik *et al.*, "Calibration and evaluation of car following models using real-world driving data," in *Proc. IEEE ITSC*. IEEE, 2017, pp. 1–6.

[200] V. Alexiadis, J. Colyar *et al.*, "The next generation simulation program," *Institute of Transportation Engineers*, vol. 74, no. 8, p. 22, 2004.

[201] R. Hamzeie, P. T. Savolainen *et al.*, "Driver speed selection and crash risk: Insights from the naturalistic driving study," *Journal of safety research*, vol. 63, pp. 187–194, 2017.

[202] D. Yang, S. Zheng *et al.*, "A dynamic lane-changing trajectory planning model for automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 228–247, 2018.

[203] S. Zhang, G. Deng *et al.*, "Optimal vehicle lane change trajectory planning in multi-vehicle traffic environments," *Applied Sciences*, vol. 12, no. 19, p. 9662, 2022.

[204] Q. Li, Z. Lei *et al.*, "An automatic conflict detection framework for urban intersections based on an improved time difference to collision indicator," *Remote Sensing*, vol. 13, no. 24, p. 4994, 2021.

[205] W. Zhao, S. Gong *et al.*, "Developing a new integrated advanced driver assistance system in a connected vehicle environment," *Expert Systems with Applications*, vol. 238, p. 121733, 2024.

[206] E. Leurent, "An environment for autonomous driving decision-making," https://github.com/eleurent/highway-env, 2018.

[207] J. Betz, A. Wischnewski *et al.*, "A software architecture for an autonomous racecar," in *Proceedings of the IEEE Vehicular Technology Conference*. IEEE, 2019, pp. 1–6.

[208] J. Betz, H. Zheng *et al.*, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 458–488, 2022.

[209] D. Caporale *et al.*, "Towards the design of robotic drivers for full-scale self-driving racing cars," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5643–5649.

[210] C. Jung, A. Finazzi *et al.*, "An autonomous system for head-to-head race: Design, implementation and analysis; team KAIST at the indy autonomous challenge," *arXiv preprint arXiv:2303.09463*, 2023.

[211] A. Raji123 *et al.*, "er. autopilot 1.0: The full autonomous stack for oval racing at high speeds."

[212] J. Betz *et al.*, "Tum autonomous motorsport: An autonomous racing software for the indy autonomous challenge," *Journal of Field Robotics*, vol. 40, no. 4, pp. 783–809, 2023.

[213] J. Kabzan, M. I. Valls *et al.*, "Amz driverless: The full autonomous racing system," *Journal of Field Robotics*, vol. 37, no. 7, pp. 1267–1294, 2020.

[214] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *International Conference on Machine Learning*, 2017, pp. 2778–2787.

[215] M. Bevilacqua, A. Tsourdos, and A. Starr, "Particle swarm for path planning in a racing circuit simulation," in *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE, 2017, pp. 1–6.

[216] S. Lovato and M. Massaro, "Three-dimensional fixed-trajectory approaches to the minimum-lap time of road vehicles," *Vehicle System Dynamics*, vol. 60, no. 11, pp. 3650–3667, 2022.

[217] S. Grollius, M. Ligges, J. Ruskowski, and A. Grabmaier, "Concept of an automotive lidar target simulator for direct time-of-flight lidar," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 825–835, 2021.

[218] C. You and P. Tsiotras, "High-speed cornering for autonomous off-road rally racing," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 485–501, 2021.

[219] F. Sauerbeck, L. Baierlein *et al.*, "A combined lidar-camera localization for autonomous race cars," *SAE International Journal of Connected and Automated Vehicles*, vol. 5, no. 12-05-01-0006, pp. 61–71, 2022.

[220] F. Massa *et al.*, "Lidar-based gnss denied localization for autonomous racing cars," *Sensors*, vol. 20, no. 14, p. 3992, 2020.

[221] F. Sauerbeck, S. Huch *et al.*, "Learn to see fast: Lessons learned from autonomous racing on how to develop perception systems," *IEEE Access*, vol. 11, pp. 44 034–44 050, 2023.

[222] K. Huang, B. Shi *et al.*, "Multi-modal sensor fusion for auto driving perception: A survey," *arXiv preprint arXiv:2202.02703*, 2022.

[223] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious NMPC with gaussian process dynamics for autonomous miniature race cars," in *Proceedings of the European Control Conference*, 2018, pp. 1341–1348.

[224] P. A. Theodosis and J. C. Gerdes, "Nonlinear optimization of a racing line for an autonomous racecar using professional driving techniques," in *Dynamic Systems and Control Conference*, vol. 45295. American Society of Mechanical Engineers, 2012, pp. 235–241.

[225] J. L. Vázquez *et al.*, "Optimization-based hierarchical motion planning for autonomous racing," in *IEEE conference on intelligent robots and systems*. IEEE, 2020, pp. 2397–2403.

[226] Y. Song, H. Lin *et al.*, "Autonomous overtaking in gran turismo sport using curriculum reinforcement learning," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2021, pp. 9403–9409.

[227] P. R. Wurman, S. Barrett *et al.*, "Outracing champion gran turismo drivers with deep reinforcement learning," *Nature*, vol. 602, no. 7896, pp. 223–228, 2022.

[228] F. Fuchs *et al.*, "Super-human performance in gran turismo sport using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4257–4264, 2021.

[229] A. Remonda, S. Krebs *et al.*, "Formula rl: Deep reinforcement learning for autonomous racing using telemetry data," *arXiv preprint arXiv:2104.11106*, 2021.

[230] J. Niu, Y. Hu *et al.*, "Two-stage safe reinforcement learning for high-speed autonomous racing," in *2020 IEEE International Conference on Systems, Man, and Cybernetics*, 2020, pp. 3934–3941.

[231] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[232] F. Tong, R. Liu *et al.*, "Multi-policy soft actor-critic reinforcement learning for autonomous racing," in *2024 IEEE 18th International Conference on Advanced Motion Control (AMC)*, 2024, pp. 1–7.

[233] L. Wang, Q. Cai, Z. Yang, and Z. Wang, "Neural policy gradient methods: Global optimality and rates of convergence," *arXiv preprint arXiv:1909.01150*, 2019.

[234] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *International conference on machine learning*. PMLR, 2019, pp. 2052–2062.

[235] C. Qi, C. Wu *et al.*, "UAV path planning based on the improved ppo algorithm," in *2022 Asia Conference on Advanced Robotics, Automation, and Control Engineering (ARACE)*, 2022, pp. 193–199.

[236] W. Chen, K. K. L. Wong, S. Long, and Z. Sun, "Relative entropy of correct proximal policy optimization algorithms with modified penalty factor in complex environment," *Entropy*, vol. 24, no. 4, p. 440, 2022.

[237] J. Bharadiya, "Convolutional neural networks for image classification," *International Journal of Innovative Science and Research Technology*, vol. 8, no. 5, pp. 673–677, 2023.

[238] F. N. Iandola, S. Han *et al.*, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[239] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv e-prints*, pp. arXiv–1707, 2017.

[240] I. Parberry, *Introduction to Game Physics with Box2D*. CRC Press, 2017.

[241] M. Estrada, S. Li, and X. Cai, "Feedback linearization of car dynamics for racing via reinforcement learning," *arXiv preprint arXiv:2110.10441*, 2021.

[242] A. Salvaji, H. Taylor, *et al.*, "Racing towards reinforcement learning based control of an autonomous formula sae car," *arXiv preprint arXiv:2308.13088*, 2023.

[243] N. A. Spielberg, M. Templer, *et al.*, "Learning policies for automated racing using vehicle model gradients," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 4, pp. 130–142, 2023.

[244] B. D. Evans *et al.*, "Safe reinforcement learning for high-speed autonomous racing," *Cognitive Robotics*, vol. 3, pp. 107–126, 2023.

[245] E. Ghignone, N. Baumann *et al.*, "Tc-driver: A trajectory conditioned reinforcement learning approach to zero-shot autonomous racing," *Field Robotics*, vol. 3, no. 1, pp. 637–651, 2023.

[246] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*. IEEE, 2018, pp. 0210–0215.

[247] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature machine intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[248] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[249] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," *Field and Service Robotics*, pp. 621–635, 2018.

[250] X. Wang, Z. Zhu, G. Huang, X. Chen, J. Zhu, and J. Lu, "Drivedreamer: Towards real-world-drive world models for autonomous driving," in *European Conference on Computer Vision*. Springer, 2024, pp. 55–72.

[251] H. Zhou, W. Li, Z. Kong, J. Guo, Y. Zhang, B. Yu, L. Zhang, and C. Liu, "Deepbillboard: Systematic physical-world testing of autonomous driving systems," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020, pp. 347–358.

[252] K. Li, K. Chen, H. Wang, L. Hong, C. Ye, J. Han, Y. Chen, W. Zhang, C. Xu, D.-Y. Yeung *et al.*, "Coda: A real-world road corner case dataset for object detection in autonomous driving," in *European Conference on Computer Vision*. Springer, 2022, pp. 406–423.

[253] Y. Pan, C. Cheng, K. Saigol, W. Lee, R. Yan, E. Theodorou, and B. Boots, "Virtual to real reinforcement learning for autonomous driving," in *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2017.

[254] M. Maramotti and A. Broggi, "Tackling real-world autonomous driving using deep reinforcement learning," in *IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 1274–1281.

[255] B. R. Kiran, Y. He, M. Alazab, A. Al-Dhelaan, M. A. Awadh, and R. Alsaqour, "A comprehensive survey of deep reinforcement learning methods and applications in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2021.

[256] B. Osiński, A. Jakubowski, P. Zięcina, P. Miłoś, C. Galias, S. Homoceanu, and H. Michalewski, "Simulation-based reinforcement learning for real-world autonomous driving," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 6411–6418.

[257] X. Fang, Q. Zhang, Y. Gao, and D. Zhao, "Offline reinforcement learning for autonomous driving with real world driving data," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 3417–3422.

[258] N. Karnchanachari, D. Geromichalos, K. S. Tan, N. Li, C. Eriksen, S. Yaghoubi, N. Mehdipour, G. Bernasconi, W. K. Fong, Y. Guo *et al.*, "Towards learning-based planning: The nuplan benchmark for real-world autonomous driving," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 629–636.

[259] G. Rossolini, F. Nesti, G. D'Amico, S. Nair, A. Biondi, and G. Buttazzo, "On the real-world adversarial robustness of real-time semantic segmentation models for autonomous driving," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[260] M. Bellone, A. Ismailogullari, J. Müür, O. Nissin, R. Sell, and R.-M. Soe, "Autonomous driving in the real-world: The weather challenge in the sohjoa baltic project," in *Towards connected and autonomous vehicle highways: Technical, security and social challenges*. Springer, 2021, pp. 229–255.

[261] H. Niu, J. Hu, Z. Cui, and Y. Zhang, "Dr^{2}l: Surfacing corner cases to robustify autonomous driving via domain randomization reinforcement learning," in *Proceedings of the 5th Int. Conf. on Computer Science and Application Engineering (CSAE)*. ACM, 2021.

[262] L. Wen, J. Duan, S. E. Li, S. Xu, and H. Peng, "Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4986–4995, 2020.

[263] A. K. et al., "Learning to drive in a day," *International Journal of Robotics Research*, vol. 38, no. 12, pp. 1370–1393, 2019.

[264] G. Wang, H. Niu, D. Zhu, and J. Hu, "Modular deep reinforcement learning for autonomous driving," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 1063–1075, 2023.

[265] Y. Li, J. Zhang, W. Zhang, and K. Li, "Deep reinforcement learning for autonomous driving based on safety experience replay," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 16 993–17 003, 2022.

[266] S. Park, H. Kim, H. Kim, and J. Choi, "Pruning with scaled policy constraints for light-weight reinforcement learning," *IEEE Access*, vol. 12, pp. 36 055–36 065, 2024.

[267] T. Fleischer, M. Puhe, J. Schippl, and Y. Yamasaki, "Public expectations regarding regulatory changes for autonomous driving," *Transportation Research Part A: Policy and Practice*, vol. 160, pp. 172–182, 2022.

[268] J.-F. Bonnefon, A. Shariff, and I. Rahwan, "The social dilemma of autonomous vehicles," *Science*, vol. 352, no. 6293, pp. 1573–1576, 2016.

[269] P. Lin, *Why Ethics Matters for Autonomous Cars*. Springer, 2016, pp. 69–85.