Towards Learning Feasible Hierarchical Decision-Making Policies in Urban Autonomous Driving

by

Mohammad Al-Sharman

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2022
© Mohammad Al-Sharman 2022

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Homayoun Najjaran Professor, Dept. of Electrical and Computer Engineering, University of Victoria
Supervisors:	Derek Rayside Professor, Dept. of Electrical and Computer Engineering, University of Waterloo
	William Melek University Research Chair Professor, Dept. of Mechanical and Mechatronics Engineering, University of Waterloo
Internal Members:	Krzysztof Czarnecki Professor, Dept. of Electrical and Computer Engineering, University of Waterloo
	Mark Crowley Associate Professor, Dept. of Electrical and Computer Engineering, University of Waterloo
Internal-External Member:	Nasser Lashgarian Azad Associate Professor, Dept. of Systems Design Engineering, University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Modern learning-based algorithms, powered by advanced deep structured neural nets, have multifacetedly facilitated automated driving platforms, spanning from scene characterization and perception to low-level control and state estimation schemes. Nonetheless, urban autonomous driving is regarded as a challenging application for machine learning (ML) and artificial intelligence (AI) since the learnt driving policies must handle complex multi-agent driving scenarios with indeterministic intentions of road participants. In the case of unsignalized intersections, automating the decision-making process at these safety-critical environments entails comprehending numerous layers of abstractions associated with learning robust driving behaviors to allow the vehicle to drive safely and efficiently.

Based on our in-depth investigation, we discern that an efficient, yet safe, decision-making scheme for navigating real-world unsignalized intersections does not exist yet. The state-of-the-art schemes lacked practicality to handle real-life complex scenarios as they utilize Low-fidelity vehicle dynamic models which makes them incapable of simulating the real dynamic motion in real-life driving applications. In addition, the conservative behavior of autonomous vehicles, which often overreact to threats which have low likelihood, degrades the overall driving quality and jeopardizes safety. Hence, enhancing driving behavior is essential to attain agile, yet safe, traversing maneuvers in such multi-agent environments. Therefore, the main goal of conducting this PhD research is to develop high-fidelity learning-based frameworks to enhance the autonomous decision-making process at these safety-critical environments.

We focus this PhD dissertation on three correlated and complementary research challenges. In our first research challenge, we conduct an in-depth and comprehensive survey on the state-of-the-art learning-based decision-making schemes with the objective of identifying the main shortcomings and potential research avenues. Based on the research directions concluded, we propose, in Problem II and Problem III, novel learning-based frameworks with the objective of enhancing safety and efficiency at different decision-making levels. In Problem II, we develop a novel sensor-independent state estimation for a safety-critical system in urban driving using deep learning techniques. A neural inference model is developed and trained via deep-learning training techniques to obtain accurate state estimates using indirect measurements of vehicle dynamic states and powertrain states. In Problem III, we propose a novel hierarchical reinforcement learning-based decision-making architecture for learning left-turn policies at four-way unsignalized intersections with feasibility guarantees. The proposed technique involves an integration of two main decision-making layers; a high-level learning-based behavioral planning layer which adopts soft actor-critic principles to learn high-level, non-conservative yet safe, driving behaviors, and a motion

planning layer that uses low-level Model Predictive Control (MPC) principles to ensure feasibility of the two-dimensional left-turn maneuver. The high-level layer generates reference signals of velocity and yaw angle for the ego vehicle taking into account safety and collision avoidance with the intersection vehicles, whereas the low-level planning layer solves an optimization problem to track these reference commands considering several vehicle dynamic constraints and ride comfort.

Acknowledgements

All glory be to Allah (God) Almighty; without his assistance and blessings, I would not have been able to continue my quest for knowledge.

Herewith, I would like to express my thanks to a lengthy list of people, without whose constant support, the work detailed in this dissertation would not have been attainable.

First and foremost, words cannot express how thankful I am to my outstanding supervisors, Prof. William Melek and Prof. Derek Rayside, for their unrelenting optimism, unending support on several levels, and enlightening arguments. Their positive attitude, approachability, and other non-academic qualities are admirable traits that I aim to adopt. Without their guidance, I would not have been able to become the person I am today.

To my internal PhD committee members, Prof. Mark Crowley, Prof. krzysztof Czarnecki, and Prof. Naser Lashgarian Azad, thank you for investing your significant time and expertise to monitoring my work, as well as for providing insightful comments and insights throughout the PhD milestones. To the external committee member, Prof. Homayoun Najjaran, I would like to convey my sincere gratitude to you for consenting to join as an external member and for devoting time and effort to reviewing the thesis.

To my father,"Khaled", mothers, "Khairieh and Asma[1]", and brothers and sisters. I consider myself tremendously fortunate to have you as an integral part in my life. Thank you for your patience and unwavering encouragement. To my wife, "Dana", and my one-year-old son "Faris (typed by him)", thank you for the love and the trust I see in your eyes.

To my WATOnomous research colleagues, Rowan Dempster, Mohamed Daoud, Anita Hu, Sinclair Hudson, Martin Ethier, Alex Zhuang, QuanQuan Li, Eddy Zhou, Kushant Patel, Yeshu Jain, and Jeffery Li, thank you for your collaboration and contribution. This work would not have been possible without you.

To my Friends in Canada, Bara Emran, Omar Sababha, Omar Farhat, Mahmoud Nasr, Parastoo Baghaei Ravari, Reza Babaee, Heba Farag, Heba Alattas, Heba El-Sawaf, Mohamed Mehrez, Omar Farhat, Hesham Jamal, Mohammad Shahab, Omar Sababha, Mohammad Mashagbeh, Ahmed Hussien, Manaf Bin-Yahya, Jamal Busnaina, Hasan J. Mrayeh, Ahmed Alquraan, I thank you all for your affection and for making my time at Waterloo more enjoyable.

To my professors in the American University of Sharjah, Prof. Mohammad Al-Jarrah, Prof. Mamoun Abdel-hafez, and Prof. Mohammad Jaradat, I want to convey my deepest

[1] in-law

thanks to you all for being so responsive and supportive. Without your direction and advice, I would not have been able to take this significant step.

To the PhD coordinator, Cassandra Brett, thank you for all your help and assistance throughout my PhD studies.

To Prof. Omar Ramahi, I would like to express my heartfelt appreciation to you for your time and support. Your words of wisdom and the experiences you have shared with me have left an immense impact on me. Chatting with you while strolling in the UW ring-road was more than enough to get me back on my feet. To my dearest brother, Milad Olaimat, I am deeply grateful for all what you have done to me. Your genuine devotion never fails to astound me.

"I am not young enough to know everything."
- Oscar Wilde

Dedication

To my beloved parents, "Khaled Al-Shorman & Khairieh Abuhmaid",
whose words of support and encouragement to perseverance echo in my ears

To my lovely wife, "Dana",
whose sincere love never ceases to amaze me

To my adorable son,"Faris",
whose spontaneous fleeting grin is a stress reliever

To my closest friends,"Murad Qasaimeh & Ahmad Altalmas",
who have never abandoned me in my rough patches

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AD Autonomous Driving

ADAM Adaptive Moment Estimation Technique

ADS Automated Driving System

ANNs Artificial Neural Networks

AVs Autonomous Vehicles

CPS Cyber Physical System

DDT Dynamic Driving Task

DNN Deep Neural Network

DRL Deep Reinforcement learning

EV Electric Vehicle

ML Machine Learning

POMDP Partially Observable Markov Decision Process

RL Reinforcement Learning

RSC Road Surface Condition

SAE Society of Automotive Engineers

TTC Time To Collision

V2V vehicle-to-vehicle

List of Symbols

S Observed States

A Set of actions

T Transition function

γ discount factor

R Set of rewards

v_{lim} Maximum urban speed limit

λ Success rate over 10 consecutive evaluations

$\mathcal{D}$ Replay buffer

π_ψ Policy that maps the states into actions

ξ Independent noise sequences

α entropy regularization coefficient

$\mathbb{R}$ Real numbers

$\dot{(\cdot)}$ Time derivative

$\mathbf{x}$ Vehicle model states

CG Center of gravity of the vehicle

δ_f Front wheel steering angle

L Distance between front axis and rear axis of the vehicle

β_s Vehicle side-slip angle

θ Yaw angle of the vehicle

l_r Distance between rear axle and CG

$\mathbf{u}$ Control actions

$\mathbf{z}$ Augmented system states

Q States error weighting matrix

R Change in control actions weighting matrix

t_0 Initial time

T_H Prediction horizon

ICR Instantaneous center of rotation

$(\cdot)_e$ Ego vehicle's state

$(\cdot)_{tar}$ Target vehicle's state

Chapter 1

Introduction

In this chapter, we first introduce the decision-making hierarchy for autonomous urban vehicles. Following that, we highlight the motivation for this PhD thesis along with the research objectives. The thesis contributions are then outlined. Finally, we provide an outline for the remainder of the thesis.

1.1 Decision Making in Autonomous Urban Vehicles

Autonomous Vehicles are considered autonomous decision-making systems as they provide continuous decisions based on processing perceptual observations. Along with these observations and sensor models, the predefined road network data, driving rules and regulations, dynamic behaviour of the vehicle, are utilized for predicting the vehicle's motion and generating low-level control commands autonomously. Developing such decision-making systems with a high degree of autonomy, is commonly organized by a well-defined multi-staged process [1].

A hierarchy of the decision-making processes of autonomous urban vehicles is depicted in Fig. 1.1. It consists of four cascaded layers, starting with the high-level route planning, followed by the behavioural path planning and motion planning layers, and the low-level feedback control completes the scheme. At the very top layer, given the predefined destination, the autonomous decision-making system must run inherent route planning algorithms to compute the optimal path using the road network as a network graph. In this layer, the edge weights are summed in order to effectively solve for the routes with minimum cost. However, as the road network becomes larger, its graph network also becomes more

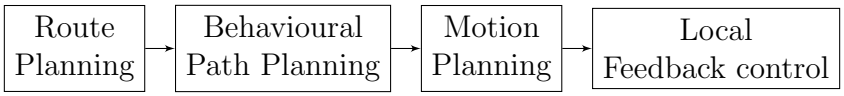


Figure 1.1: Decision-making processes in urban autonomous vehicles.

complex making the use of the classical route planning schemes, namely Dijkstra [2] and A* [3], inefficient as the search time may exceed seconds [4]. Intelligent route planning approaches have been introduced for transportation efficiency enhancement; advanced Deep Learning and Internet of Things (IoT) technologies have been employed for efficient route planning in complex urban transportation [5, 6]. Once the optimal route has been defined, the next layer is focused on behavioral path planning. This layer is responsible for choosing proper driving behaviour based on the observed behaviour of other road drivers, traffic signals and road surface conditions. This behavior enables the AV to interact with road participants while performing lane changing, lane following, and other more complicated tasks like intersection-traversal. For instance, choosing a cautious behavior for intersection-traversal maneuvers based on the road conditions is a responsibility of the behavioral path planner. After the driving behavior has been determined by the behavioral path planner, this behaviour has to be mapped into a vehicle's trajectory which will be tracked by the local embedded controller. Different aspects must be taken into account while choosing the proper trajectory (e.g. it must be feasible taking into consideration the vehicle's dynamic model constraints, the guarantee of ride comfort and safety). Finding such trajectory is an inherent component that must be accounted for by the motion planning layer. Finally, to execute this trajectory, a feedback controller must be tuned to provide the correct input to govern the planned motion and compensate for the tracking errors arising from the assumptions made on the utilized vehicle dynamic model [7].

Besides the scene complexity, making decisions which lead to efficient and safe transportation in urban driving settings, such as unsignalized intersections, is an entangled task due to several factors: i) restricted sensing capabilities, specifically, vision and proximity in such time-varying environment; ii) cluttering and occlusions in the scene which impede achieving accurate perception; iii) legal and technical constraints on the vehicle's response, arising from the driving rules and regulations. Hence, the primary motivation of this research is to provide more efficient learning-based solutions for enhancing decision making, at several levels; starting from low-level state estimation and control to high-level behavioral planning, of autonomous vehicles in urban environments.

1.2 Motivation

Following the Defense Advanced Research Projects Agency (DARPA) Urban Driving Challenge (UDC), held in 2007, the research community has been encouraged to develop novel technologies to address technical and social challenges concomitantly with driving in urban settings autonomously [8, 9, 10]. These challenges stem from the nature of urban driving itself, characterized by its complex multi-agent motion planning, in which the vehicle must react to various different scenarios including the interaction with other vehicles and traffic signals and signs [11, 12, 13]. Unlike highway autonomous Driving [14, 15], driving in urban environments requires effective handling of complex multi-agent scenarios with a high level of uncertainty and occlusions [16, 17, 18]. More specifically, driving at intersections is considered perilous for most human drivers. This can be justified by looking at the data reported in [19]. The Fatality Analysis Reporting System (FARS) and National Automotive Sampling System General Estimates System (NASS-GES) provide an estimation that 40% of the crashes recorded in the US in 2008 occurred at intersections. They reasoned that among the factors contributing to these crashes, the most prevalent were related to the crash-involved drivers, namely, their age, sex and driving behaviour. Hence, deriving safe policies for autonomous vehicles that allow for safe crossing behaviour at intersections has been a topic of a profound importance as it can provide useful guidelines for designing preventive crash-mitigation schemes. Recently, academia and industrial partners have been extensively testing the most advanced autonomous technologies on their platform to ensure safe and efficient urban driving [20, 21, 22]. However, within the context of urban intersections, enabling an autonomous vehicle to perform dynamic tasks and navigate safely and efficiently in such complex urban environments requires high degree of autonomy, according to the Society of Automotive Engineers (SAE) J3016 standard [23]. Nonetheless, the current automated vehicles, even the fully autonomous ones, cannot fully navigate safely at all times, and cannot guarantee crash-free maneuvers due to critical decision-making errors [24].

Making decisions at urban unsignalized intersections is a highly intractable process. The complex driving behaviour and the disappearance of traffic control signals makes the motion inference of other intersection users highly-challenging [25, 26, 27]. The non-stationarity problem, along with the large partially-observable state space of agents dictate designing robust algorithms for safe intersection-traversal [28]. Numerous studies have investigated motion planning and decision-making algorithms to enhance the driving safety at unsignalized intersections. These algorithms have been introduced to tackle two main problems; inferring the intention of other intersection users and planning the ego vehicle's motion while traversing the intersection [29]. While this PhD thesis is primarily concerned

with the latter problem, in **Chapter 3**, we shed light on the main state-of-the-art intention inference schemes along with recommendations for enhanced driver intention predictions.

Recently, urban decision-making using advanced learning-based algorithms has been the focus for numerous research projects [30, 31, 32]. However, these algorithms lacked practicality to handle real-life complex scenarios for multiple reasons: First, most of these schemes utilize Low-fidelity vehicle dynamic models which makes them incapable of simulating the real dynamic motion in real-life driving applications [33, 34, 35, 36]; Second, these learning-based schemes are trained and tested via experiments conducted in a simulated environment which lacks realizations of the actual driving environment [37, 38]. Hence, the main goal of this research is to develop high-fidelity learning-based frameworks to enhance the autonomous decision-making process at these safety-critical environments. Therefore, our first research challenge is to conduct an in-depth and comprehensive survey on the state-of-the-art learning-based decision-making schemes with the objective of identifying the main shortcomings and potential research avenues. Taking these directions into account, in our second and third research problems, discussed in **Chapter 4** and **Chapter 5**, we propose novel learning-based frameworks with the objective of enhancing saftey and efficiency at different decision-making levels. These research problems include developing a low-level sensor-independent state estimation technique for a safety-critical cyber-physical system and feasible multi-layer decision-making in urban environments, respectively.

1.2.1 Learning-Based Decision-Making at Urban Unsignalized Intersections: A Survey

Decision-making algorithms, in this survey, can be classified into three main categories: cooperative approaches, including game-theoretic, heuristic-based approaches and hybrid approaches which combine multiple classes of these algorithms for handling the unsignalized intersection problem. Cooperative approaches entail the use of vehicle-to-vehicle (V2V) communication technology to exchange the states between the subject vehicle and other intersections users [39, 40, 41]. However, such technology is still an active area of research and has not been sufficiently developed to allow its application in existing decision-making schemes. Game-Theoretic-Based algorithms were adopted to model the vehicles' interactions in unsignalised intersections [42, 43]. These game-theoretic based approaches assume that the states of the interacting vehicles are observed by the subject vehicle, which allows for predicting their future trajectories and then plan its own. However, this assumption is not likely to hold for current real-life decision making at unsignalized intersections. Heuristics-based approaches have been engineered to tackle safety-oriented problems asso-

ciated with traversing urban intersections [44]. Researchers commonly classify these approaches into two main groups: rule-based and ML approaches [30]. Rule-based approaches use safety intersection metrics, namely TTC, to generate distance-based traversing rules. However, engineering such rules to adapt with various possible crossing situations is a tedious process due to the large number of rules which need to be tuned. ML-based approaches, especially RL approaches, focus on learning driving policies from the interaction between the vehicle and the intersection environment.

Applying modern RL-based approaches for approximating optimal driving policies at unsignalized intersections has been studied extensively in the literature. Researchers have been motivated to develop these algorithms, owing to their capabilities in handling partially-observable environments by training its data-driven models based on mapping the environmental observations into actions [45]. Nevertheless, design challenges behind developing crash-free intersection maneuvers and deploying them in real driving environments still need to be overcome. The surveyed schemes still suffer from several problems,i.e, the proposed design assumptions, the scalability of the proposed scheme to deal with more challenging urban driving scenarios, and the experimental validation in real urban driving settings. Hence, motivated by the published works, a review of the current and emerging trends in aspects related to decision-making in urban unsignalized intersections is recommended to lay the groundwork for potential advancement in this research direction. Thus, in **Chapter 3**, we offer an overview of algorithms and applications of decision-making in urban autonomous vehicles at unsignalized intersections, with the goal of identifying knowledge gaps in this literature and introduce our contributions, presented in **Chapter 4** and **Chapter 5**, to enable safe and effective decision-making in the autonomous driving scenarios this thesis is focusing on.

1.2.2 Cyber-physical System State Estimation in Urban Driving

Since the focus of this PhD dissertation is to develop safety-critical decision-making algorithms, accurate observations of the driving environment, including the vehicle's cyber-physical safety-critical systems, such as braking [46, 47, 48], is required. With increased autonomy and control authority, however, it becomes increasingly pivotal that the braking system be accurate and safe against faults. Braking control generally uses measurements of the hydraulic pressure in brakes to decide actions to be taken, measured by pressure sensors [49]. If a hardware or software fault occurs in these sensors, however, brake control can be compromised, leading to potentially dangerous safety issues. This can be circumvented by developing data-driven brake pressure state estimation technique using indirect

measurements, with the potential to evolve the system into a sensor-independent system with sufficiently accurate estimation [50].

Obtaining highly expressive state estimator's model is linked in practice to the training method utilized. Despite the effectiveness of conventional training techniques, modern deep-learning-based structures have shown superiority in terms of improving the associated overfitting and achieving significant generalization capabilities. In [51], a neural net was utilized to perform estimation of brake pressure, using data obtained from an electric vehicles (EV). A conventional back-propagation is adopted for training the ANN-based state estimator. However, conventional back-propagation suffers from problems, such as overfitting and vanishing gradient, as well as higher computational burden in training. Nevertheless, these problems have been resolved by implementing the recent advances of Deep Learning techniques to augment the training process of the deep neural network (DNN) [52]. Inspired by these significant features, in **Chapter 4**, a DNN is introduced and trained using deep-learning training techniques to infer state estimates of a safety-critical system with high-accuracy.

1.2.3 Hierarchical Reinforced-learning for Feasible Decision-Making

Deep Reinforcement Learning (DRL) techniques have been employed for deriving safe driving policies at unsignalized intersections for autonomous vehicles due to their significant capabilities in handling high-dimensional perceptual observations with discrete and continuous action spaces [53, 54]. However, the more complicated the driving scenario, the more training time is required for the DRL algorithm to converge. Hence, for learning complex behaviors efficiently with less training iterations, Curriculum Learning (CL) principles were applied while training an autonomous agent. CL was proposed in [55] as a way to accelerate learning by first training the system on simple tasks and thereafter progressively increasing the difficulty of the tasks given to the learning agent. Apart from learning high-speed autonomous overtaking [56], learning through designed curricula has also manifested significant learning benefits in terms of training time reduction and faster convergence in urban driving settings at unsignalized intersections. For instance, in [30], a curriculum DRL-based motion planning system was proposed for crossing a four-way unsignalized intersection autonomously. The proposed algorithm was designed to generate curricula in order to learn the crossing policy with fewer training iterations. However, simple one-dimensional crossing behavior is learned, while other more complex scenarios, such as two-dimensional left-turn was not investigated.

Designing autonomous left-turn decision-making frameworks at unsignalized intersections is deemed a challenging engineering problem as intersection-users turning behaviors

are not governed by traffic control signals [57]. This problem can be modeled as a Markov Decision Process (MDP) where solutions can be obtained by utilizing online solvers or through optimal policy approximation using RL approaches. For instance, in [57, 58], the Adaptive Belief Tree (ABT) solver is used to solve the formulated Partially-Observable Markov Decision Process (POMDP) [59] of the decision-making problem. Authors use the *Critical-Turning-Point (CTP)* approach where the left turn trajectory is simply assumed as a straight line with a quarter circle curve. However, online solvers work only for fairly small state spaces, and for larger state spaces the complexity of solving MDP scales dramatically. On the other hand, Deep Reinforcement Learning (DRL), can work with much larger, or even continuous spaces, such as Atari [60]. Furthermore, DRL approaches can also approximate their optimal policies without the requirement of observing the full state space. Considering these qualities of DRL architectures, [61, 62] introduced reinforcement-learning-based frameworks to generate safe driving policies for left-turn. However, On the behavioral planning side, the utilization of Deep Q-network (DQN) method is inefficient for urban driving environment which requires continuous actions rather than discrete ones. On the low-level side, similar to Stanley and Pure Pursuit controllers, they used geometric controller does not represent actual vehicle constraints, e.g. max steering rate, although it can minimize the tracking error. In addition, as the error does not consider error dynamics over time, their performance deteriorates significantly at high speeds which makes them suitable only for low-speed maneuvers.

Numerous research papers have addressed the low-level motion planning problem and control at urban unsignalized intersections using Model Predictive Control (MPC) principles. For instance, Hu *et al.* [63] proposed an event-triggered model predictive adaptive dynamic programming technique for motion planning at urban intersections. The method takes urban speed, vehicle kinematics and road constraints into consideration while solving a multi-objective optimization problem. However, for high-fidelity decision-making applications in urban autonomous driving, incorporating the local motion planning and low-level control layers and taking into account vehicle dynamics is essential to ensure the feasibility of the high-level RL commanded actions. Such integration has been attempted for intersection-management applications, where centralized reference signals being distributed to the intersections agents via V2V communication [64, 65]. In [65], an integration between high-level decision-making layers and low-level MPC-based motion planing layer has been proposed for learning supervisory intersection-management policy in connected driving fashion. However, as far as we know, such integration has not been developed for learning intersection-traversal policy of the ego vehicle agent. Hence, having the motion planing layer integrated while approximating, non-conservative yet safe [66], intersection-traversal policies would facilitate learning, with feasibility guarantees [67], taking into

account lateral and longitudinal dynamics.

1.3 Research Contributions

In this section, we show the research contributions for proposed research problems.

1.3.1 Learning-Based Decision-Making at Urban Unsignalized Intersections: A Survey

In **Chapter 3**, we direct our attention towards various aspects related to behavioral motion planning for autonomous vehicles at unsignalized intersections. To be more specific, we focus this chapter on learning-based decision making schemes with a greater attention to algorithms that combine the recent advances of RL and deep learning for learning driving policies at unsignalized intersections. However, decision-making based on imitation learning or V2V communication, in a connected driving fashion, is out the of proposed Chapter's scope. Using V2V communications [68, 69] can be a potential solution for anticipating vehicle behaviour and transferring it to the ego vehicle. However, for this solution to be fully viable, vehicular communication and connected vehicle technologies must be widely deployed. It should be noted that the vehicle-pedestrian interaction behavior is not covered in this proposed research work.

The main novelty of this research work, can be stated as follows:

- an organised and in-depth state-of-the-art literature survey for decision-making at unsignalized intersections is proposed, highlighting the main navigational challenges and cutting-edge learning-based solutions.
- an exploration of the Driver Intention Inference (DII) schemes at unsignalized intersections is carried out, with the goal of identifying key remarks for better handling the large partially-observable state space of the problem.
- based on the in-depth investigation, limitations of the published learning-based decision-making frameworks are identified and potential research directions are suggested to achieve better generalization characteristics of the trained traversing policies in real-life driving scenarios. Some of these research directions have been followed in **Chapter 3** and **Chapter 4**.

This work is associated with the following journal submission:

Al-Sharman, M., Melek, W., & Rayside, D., (2022). Autonomous Driving at Unsignalized Intersection: A Review of Decision-Making Challenges and Reinforcement Learning-Based Solutions. IEEE Transactions on Vehicular Technology (under review).

1.3.2 Cyber-physical System State Estimation in Urban Driving

In **Chapter 3**, we propose a novel deep learning-based training technique for a sensor-independent safety-critical system state estimation in urban driving settings. The proposed learning scheme uses the dynamic vehicle states and powertrain states as inputs and the ground truth of the brake pressure values are the outputs while training the neural network. Compared with the conventional training technique, the proposed model has resulted in improved estimation accuracy.

The main novel contributions of the proposed state estimation algorithm can be summarized as follows:

- A novel sensor-independent deep-learning-based algorithm is developed for brake pressure state estimation of an electric vehicle;
- Compared with conventional training methods [51], the proposed approach demonstrates more accurate brake pressure state estimation with RMSE errors of 0.048 MPa;
- The proposed deep learning structure is expandable, hence, it can estimate other EV states in urban and high-way environments.

This work is associated with the following journal publication [70]:

Al-Sharman, M., Murdoch, D., Cao, D., Lv, C., Zweiri, Y., Rayside, D., & Melek, W. (2020). A Sensorless State Estimation for A Safety-Oriented Cyber-Physical System in Urban Driving: Deep Learning Approach. IEEE/CAA Journal of Automatica Sinica, 8(1), 169-178.

1.3.3 Hierarchical Reinforced-learning for Feasible decision-making

Taking on the concluded research directions obtained from Problem I, in **Chapter 5**, we emphasize that the state-of-the-art decision-making approaches focus on advancing the

high-level behavioral reasoning neglecting the importance of feasibility guarantees provided by motion planning and low-level feedback control layer [7]. We illustrate, in **Chapter 3**, that low-level control integration is required to obtain efficient policies for driving in safety-critical environments [22]. Motivated by the aforementioned features of DRL architectures, particularly the soft-actor-critic (SAC) architecture [71], which has demonstrated remarkable ability in learning driving policies for overtaking and maneuvering at roundabouts [72]. In **Chapter 5**, we propose a novel hierarchical learning-based technique for left-turn decision-making at unsignalized intersections. The Chapter offers the following main contributions:

- A novel hierarchical soft-actor-critic reinforcement learning framework is proposed, in which an integration between the behavioral planning and motion planning layers is developed to learn feasible driving policies for left-turn maneuvers at unsignalized intersection.
- High-fidelity driving policies are being trained while accounting for real-world constraints including vehicle dynamic constraints and ride comfort.
- RL baselines comparison is conducted, in which we carry out several urban driving simulation experiments to evaluate the performance of the proposed integration with other model-free algorithms.

The work of this Chapter is associated with the following journal submission:

Al-Sharman, M., R. Dempster, Rayside, D., & Melek, W. (2022). Self-Learned Autonomous Driving at Unsignalized Intersections: A Hierarchical Reinforced Learning Approach for Feasible Decision-Making. IEEE Transactions on Intelligent Transportation Systems [73].

1.4 PhD Scholarly Contributions

We would like to emphasise that, at the time of writing this thesis, multiple research contributions were published as part of my PhD thesis. Non-thesis published conferences were produced as part of WATOnomous research projects where I led several research projects for graduate and undergraduate students [74, 75, 76].

1. **Al-Sharman, M.**, Murdoch, D., Cao, D., Lv, C., Zweiri, Y., Rayside, D., & Melek, W. (2020). A sensorless state estimation for a safety-oriented cyber-physical system

in urban driving: deep learning approach. IEEE/CAA Journal of Automatica Sinica, 8(1), 169-178.

2. **Al-Sharman, M.**, Melek, W., & Rayside, D. (2022). Autonomous Driving at Unsignalized Intersection: A Review of Reinforcement Learning-based Decision-Making. IEEE Transactions on Vehicular Technology (under review).

3. **Al-Sharman, M.**, Melek, W., & Rayside, D. (2022). Self-Learned Autonomous Driving at Unsignalized Intersections: A Hierarchical Reinforced Learning Approach for Feasible Decision-Making. IEEE Transactions on Intelligent Transportation Systems (under review).

4. Rowan Dempster, **Al-Sharman, M.**, Yeshu Jain, Jeffery Li, Derek Rayside, & William Melek. In 2022 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4913-4919). IEEE.

5. Hu, C., Hudson, S., Ethier, M., **Al-Sharman, M.**, Rayside, D., & Melek, W. (2022). Sim-to-Real Domain Adaptation for Lane Detection and Classification in Autonomous Driving. In 2022 IEEE Intelligent Vehicles Symposium (IV).

1.5 Thesis Outline

We now present an outline for the remainder of the thesis. In **Chapter 2**, we briefly present background preliminaries of the research studies addressed in this thesis. We provide fundamental details of deep neural networks, model-free reinforcement learning, and longitudinal vehicle dynamics. In **Chapter 3**, we present a comprehensive survey of the learning-based decision-making schemes. We highlight the recent advances made in aspects related to the decision-making problem at unsignalized intersections. Concluded remarks and suggested research directions are then illustrated. In **Chapter 4**, our second research problem is presented. We address the problem of developing a sensor-independent state estimation for a safety-critical cyber-physical system in urban driving settings. Then, in **Chapter 5**, we demonstrate the development of an integrated behavioral planning and motion planning layers for navigation unsignalized intersections with feasibility guarantees. Finally, a summary of results and a discussion about the limitations of our decision-making approach and future directions are provided in **Chapter 6**.

Chapter 2

Background

2.1 Deep Neural Network

A deep neural network is chosen in this study to perform brake pressure state estimation. The basic architecture design of the Multilayer Neural Network which is composed of a single input layer, one or more hidden layers and a single output layer.

The elements of the input vector $I = [i_1, i_2, \ldots, i_k]$ are weighted by the weight matrix W and then summed with the neuron bias b to yield the net input n.

$$n = \sum_{i=1}^{k} w_j i_j + b \tag{2.1}$$

Then the neuron output a is generated using an activation function f.

$$a = f(n) \tag{2.2}$$

2.2 Decision-making modeling as a MDP

Several research works have envisaged the decision-making problem at intersections as a reinforcement learning problem, where the agent and the environment interact continuously to learn an optimal policy that governs the vehicles' motion. The agent takes an action and the environment responds to this action and present new scenarios to the agent. A Markov

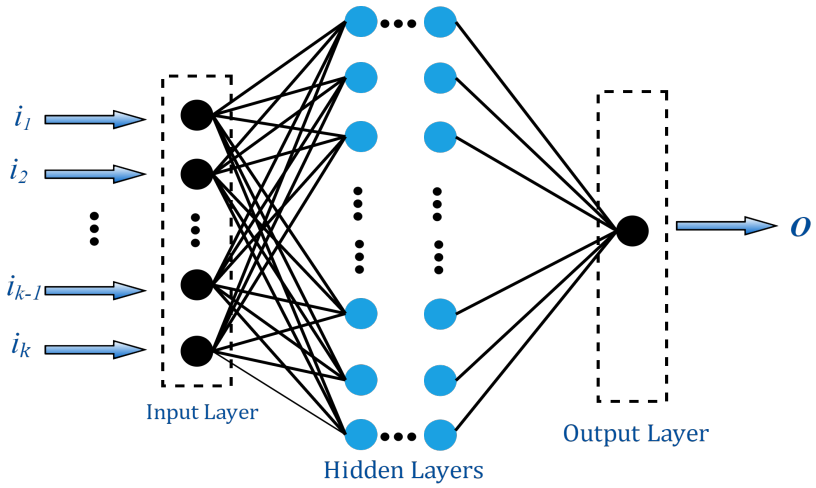


Figure 2.1: Dropout Training Technique

Decision Process (MDP) is used to describe the environment for the RL problem [30], where we assume that the environment for this specific problem is fully observable. Technically, MDP is described as a tuple $\{S, A, R, T, \gamma\}$ in which S represents the observed states. These states may include information about the ego vehicle and states of other vehicles crossing the intersection. Among these states, velocities, position, and states related to the geometry of the intersection. A and T represents the set of actions and the transition function that maps state-action pairs to a new state. The immediate reward is defined by the reward function R, whereas γ represents the discount factor for long-term rewards.

In occluded intersections where the environment is not fully observed due to limited sensor range, occlusions in the scene, or uncertainty related to the pedestrians/ drivers intentions, a Partially-Observable Markov Decision Process (POMDP) is adopted to model these types of intersections. These cases shall be discussed in more detail in **Chapter 3**.

2.3 Safety Assessment at Intersections

At high-level decision-making, drivers perform safety assessment to avoid crashes and potential hazards. Shirazi *et al.* [77] introduces five topics pertaining the safety assessment at intersections: Gap, Threat, Risk, Conflict and Accident. Gap assessment is an estimate used to anticipate the free distance between the leading and trailing vehicles. Gap distance-based and time-to-collision (TTC) algorithms have been proposed for traversing intersection [78, 79]. However, these simple approaches require laborious parameter-tuning

to deal with different intersections scenarios. Given the locations of the subject vehicle, a threat assessment process is usually conducted to anticipate the potential threats of other road participants [80]. In [81], by inferring the intention of the road participants, threat predictions were obtained using random decision trees and particle filtering. A survey on the threat assessment technologies and state-of-the-art approaches can be found in [82]. A risk assessment approach is used to detect risky scenarios which are related to the limited capabilities of the perception sensors or occluded environment which may result in incorrect decisions [83]. Risk assessment is usually coupled with predictions about the intention of other road participants. Intention-aware risk assessment has been done extensively to evaluate maneuvers at occluded intersections with limited perception capabilities. For detailed risk assessment at unsignalized intersection, the reader is referred to Section 3.1.2 which describes the state-of-the-art approaches of risk assessment for decision-making at urban intersections. Based on the environmental observations collected, conflict assessment is concerned with predicting the potential conflict scenarios of two or more vehicles that are going to collide if their movements remained unchanged [29]. Lastly, accident assessment is based on conducting precise analysis using data mining and machine learning techniques to make predictions that help in preventing crashes [84].

2.3.1 RL Approaches

Preliminaries

Reinforcement learning is a group of algorithms that focus at learning optimal policies via performing iterative experiments and evaluations for the sake of self-teaching overtime to achieve a specific goal. RL can be distinguished from other learning techniques such as supervised learning because the labels are timely delayed. The aim of RL is to learn an optimal policy π which in charge of mapping the system states to control inputs that can maximize the expected reward $J(\pi)$. In eq. (2.3), the reward r_t indicates how successful the agent was at a given time step t. For instance, large r_t values are given when the agent is close to the desired trajectory, while small r_t values are given when large deviations occur [85]. The discounted accumulated reward is given as

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi\right] \tag{2.3}$$

The discount factor γ, where $\gamma \in [0, 1]$, is used to adjust whether the agent is far-sighted

or short-sighted. The desired policy can be described as

$$\pi^* = \arg\max_{\pi} J(\pi) \tag{2.4}$$

The value of the state x, is evaluated by calculating the expected return starting from x and, subsequently, governed by policy π

$$V^{\pi}(\boldsymbol{x}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \boldsymbol{x}_0 = \boldsymbol{x}, \pi\right] \tag{2.5}$$

where $V^{\pi}(x_t)$ is defined by [86] as the *value function*. Similarly, the action value in state x is evaluated by calculating the expected reward starting from the action u in a state x and, subsequently, following policy π

$$Q^{\pi}(\boldsymbol{x}, \boldsymbol{u}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \boldsymbol{x}_0 = \boldsymbol{x}, \boldsymbol{u}_0 = \boldsymbol{u}, \pi\right] \tag{2.6}$$

where $Q^{\pi}(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is defined as the *action-value function*.

Modern machine learning-based algorithms, powered by advanced deep learning, have leveraged the use of RL principles in compact forms called Deep Reinforcement Learning (DRL) [87, 88]. These techniques have been used in the area of decision-making and motion planning for autonomous vehicles due to their significant capabilities in handling high-dimensional sensor's observations with discrete and continuous action spaces [53, 54]. In the following subsection, we will discuss several variants of DRL and their applications in navigating urban intersections.

RL Techniques

In the current literature, the problem of navigating an unsignalized intersection has been modeled as a MDP. Model-free RL learners are employed to sample the MDP to infer information about the unknown model. Numerous variants based on *Monte-Carlo* (MC) and *Temporal-difference* (TD) schemes were utilized for learning optimal policy for traversing intersections. In this review, we surveyed these approaches and their corresponding applications with greater details in the following sections.

Monte-Carlo Approaches. These learning methods can be grouped into on-policy or off-policy based on the updates that are conducted by the same policy or a different policy.

The on-policy approach uses the same policy for policy value estimation and control. On the other hand, off-policy algorithms use two separate policies; the behaviour policy and the target policy for both behaviour generation and learning the rewards corresponding to its actions, respectively [89]. An advantage of this uncoupling between the two policies is that the target policy can be deterministic (greedy), whereas the behaviour policy samples all probable actions. For convergence guarantees, two essential assumptions are made by the MC methods; first, the generated episodes must be large and, second, the states and actions have to be explored sufficiently.

Temporal-Difference Approaches. Unlike Monte-Carlo learning schemes, TD approaches make faster updates within the episode itself by using the Bellman equation. This is advantageous because it can provide faster convergence characteristics. Both MC and TD learning schemes adopt tabular methods for storing value functions of the states or the state-action pairs which make them sample inefficient when it comes to handling problems with large state space. As an example of these value based approaches, Deep Q-networks, and several of its variants, were applied to discrete state-action problems [90]. Hence, Actor-Critic approaches were designed to deal with these spaces. Actor-critic (AC) algorithms implement both the value-based approaches and the policy-based approaches. It comprises a couple of estimators: the actor network estimator which is based on Q-value, whereas the critic network utilizes the state-value function estimation. Whilst the agent's behaviour is controlled by the actor using the policy, the action is evaluated based on the value function. Deep learning-based actors and critics are introduced as DNN for function approximation purposes. A Deep deterministic policy gradient (DDPG) scheme is an example of tuning both actor and critic deep neural networks for continuous action space problems.

Due to its ability in handling large continuous action space, DDPG has been employed for learning traversing behaviours at unsignalized intersections. DDPG is a model-free, off-policy actor-critic algorithm [91, 92], which combines the actor-critic feature from the Deterministic Policy Gradient (DPG) with the target neural network and replay buffer shuffling from the DQN. With the actor-critic component, DDPG is allowed to work with the continuous action domain while learning a deterministic policy. Whereas, the experience replay stabilizes the learning of the Q-function. However, the exploration in learning in a continuous action is considered challenging. Therefore, for better exploration in DDPG, noise sequences $\mathcal{N}$ for the exploration policy $\mu'$is formulated by adding noise sequences $\mathcal{N}$ as follows.

$$\mu'\left(s_t\right) = \mu\left(s_t \mid \theta_t^{\mu}\right) + \mathcal{N} \tag{2.7}$$

Furthermore, DDPG performs soft updates on the parameters that tune the actor and

Table 2.1: Comparison of the surveyed model-free RL schemes used in intersection navigation problem

RL Approach	Features	Limitations
MC	• MC methods are *model-free*. Hence, information of transition probabilities is not required. • *Off-policy* variants of MC are Simple to design, whereas *On-policy* methods own better stability characteristics when integrated with a function approximator i.e. (Neural networks [93]).	• Slower updating process. MC method normally waits until the episode finishes to update V(s) and Q(s). As a result, they converge slower compared to TD methods. • Similar to TD, MC uses tabular method to store the value function of states which makes them ill-suited for handling complex problems (i.e multi-agent autonomous driving problems).
TD	• Similar to MC, TD methods are also *model-free* RL methods. • Faster updates. Compared to MC methods, TD performs updates using Bellman Formula at every step within the episode.	• Like MC methods, TD methods are inefficient for dealing with large and complex state spaces due to the lack of memory used for storing the value function of state-action pairs.

critic neural networks, $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$ with $\tau \ll 1$. This means that the target values are constrained and change slowly which improves the stability of learning. DDPG has also adopted the *batch normalization* feature from deep learning to resolve the problem of learning from low dimensional feature vector observations that have various physical units (i.e. acceleration vs angular velocity). This technique performs normalization for each dimension across the feature vector in a minibatch to have unit mean and variance. This can facilitate the learning process by finding the proper hyper-parameters which provide better generalization across the vehicle dynamic states that have diverse scales and units. In section 3.1.3, we illustrate the utilization of model-free DRL techniques, including DDPG and other more recent actor-critic algorithms, for learning crossing policies in continuous action spaces.

2.4 Braking Performance

This section illustrates the braking dynamics and its basic equations which are needed for understanding the vehicle's deceleration profile and corresponding stopping distances. Using Newton's Second Law, the general equation of the braking performance is given as [94]

$$Ma_x = \frac{W}{g}\left(-D_X\right) = -F_{Xf} - F_{Xr} - D_A - W\sin\theta_{up} \tag{2.8}$$

where W, g and D_X denote the vehicle's weight, gravitational acceleration and linear deceleration, respectively, F_{Xf} and F_{Xr} denote the front and rear axle braking forces, respectively, D_A and θ_{up} represent the aerodynamic drag and the uphill grade, respectively.

2.4.1 Deceleration and Stopping Distance

The braking force terms are created by the applied braking torque along with the rolling resistance, internal driveline drag and bearing friction effects. Assuming that the forces acting on the vehicle are constant while braking, the fundamental deceleration relationship is obtained as

$$D_X = -\frac{F_{Xt}}{M} = -\frac{dV}{dt} \tag{2.9}$$

where F_{Xt} denotes the total longitudinal deceleration forces and V is forward velocity.

By integrating (2.9), the velocity change can be evaluated as

$$\int_{V_o}^{V_f} dV = -\frac{F_{Xt}}{M}\int_0^{t_s} dt \tag{2.10}$$

where t_s is the time to stop.

From (2.10), in case of full stop (V_f=0), the distance traveled during the deceleration is given as:

$$SD = \frac{V_o^2}{2\frac{F_{Xt}}{M}} = \frac{V_o^2}{2D_X} \tag{2.11}$$

and the time to stop is

$$t_s = \frac{V_o^2}{2\frac{F_{Xt}}{M}} = \frac{V_o}{2D_X} \tag{2.12}$$

It can be concluded from (2.11) and (2.12) that the distance is proportional to the velocity squared, whereas the time to stop is proportional to the velocity. Taking into account the wind resistance while calculating SD, the aerodynamic drag will be involved as it depends on the square of the speed and the vehicle drag factor

$$\int_0^{SD} dX = \mathrm{M} \int_{V_o}^{0} \frac{\mathrm{VdV}}{F_b + CV^2} \tag{2.13}$$

where C is the vehicle drag factor and F_b is the total brake force acting on rear and front wheels.

$$SD = \frac{M}{2C} \ln \left[\frac{F_b + CV_o^2}{F_b} \right] \tag{2.14}$$

2.4.2 Braking Forces

The braking torque is applied to generate a braking force on the surface of the ground to decelerate the driveline and the wheel. This force is expressed as

$$F_b = \frac{T_b - I_w \alpha_w}{r} \tag{2.15}$$

where α_w and I_w denote the rotational deceleration and the rotational inertia of the vehicle.

The consistent performance of the braking torque results in consistent deceleration in braking maneuvers leading to steady stopping distances. Disc brakes have shown superiority over the Drum brakes in terms of consistent torque properties during the stop. Drum brakes show a "sag" during the intermediate region of the stop [94].

2.5 Longitudinal Vehicle Dynamics

The longitudinal vehicle dynamic model is comprised of two major dynamic models: the vehicle dynamics and the powertrain dynamics. The vehicle dynamics are affected by the forces of rolling resistances, longitudinal tire forces, aerodynamic drag forces and other gravitational forces.

$$m\ddot{x} = F_{Xf} + F_{Xr} - F_{aero} - R_{xf} - R_{xr} - w\sin\theta_{up} \tag{2.16}$$

where F_{Xf} and F_{Xr} denote the Front and rear axle longitudinal forces, respectively, F_{aero} denotes the longitudinal aerodynamic drag force, R_{Xf} and R_{Xr} represent the rolling resistance for the front and the rear tires, respectively.

Longitudinal Tire Forces

The longitudinal tire forces represent the friction forces of ground acting on the tires. These forces depend on the friction coefficient μ, slip ratio and the vertical (normal) load on the tire.

Slip Ratio The longitudinal slip is the difference between the actual longitudinal velocity at the axle of the wheel V_x and the rotational velocity of the tire $r_{eff}\ \omega_w$. The longitudinal slip ratio is described under braking as

$$\sigma_x = \frac{r_{eff}\omega_w - V_x}{V_x} \tag{2.17}$$

and during the acceleration can also be described as

$$\sigma_x = \frac{r_{eff}\omega_w - V_x}{r_{eff}\omega_w} \tag{2.18}$$

Assuming that the friction coefficient of the road-tire interaction to be 1 and the vertical normal force is constant, the longitudinal tire force can be described as a function of the slip ratio as illustrated in Fig. 2.2.

As seen in Fig. 2.2, at small slip ratio (< 0.1), the longitudinal tire force is proportional to the slip ratio. Hence, the longitudinal tire force can be modeled as

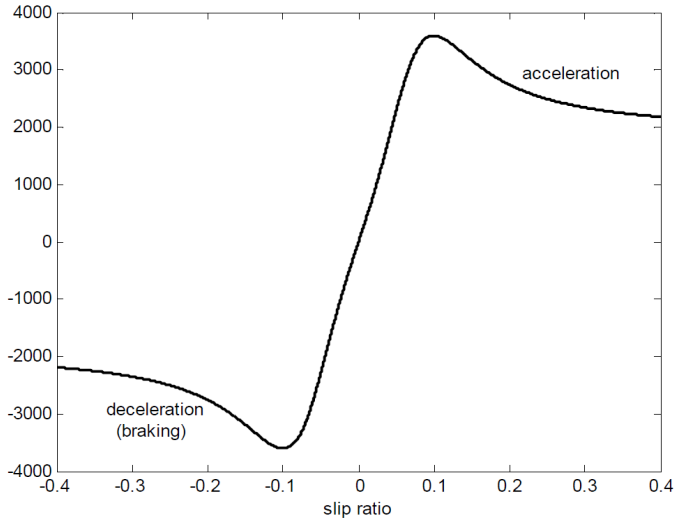


Figure 2.2: Longitudinal tire force as a function of slip ratio [94].

$$F_{xf} = C_{\sigma f}\sigma_{xf} \tag{2.19}$$

$$F_{xr} = C_{\sigma r}\sigma_{xr} \tag{2.20}$$

Where $C_{\sigma f}$ and $C_{\sigma r}$ define the longitudinal tire stiffness parameters for the front and rear tires, respectively.

At large slip ratio or if the road is slippery, a nonlinear tire model needs to be utilized to calculate the longitudinal tire force. For instance, "the magic formula" model or the tire model can be used to model the tire forces in this case.

In the proposed decision making scheme, we will include relevant longitudinal dynamic stats along with the states that are required for slip ratio calculation while the vehicle is approaching the intersection.

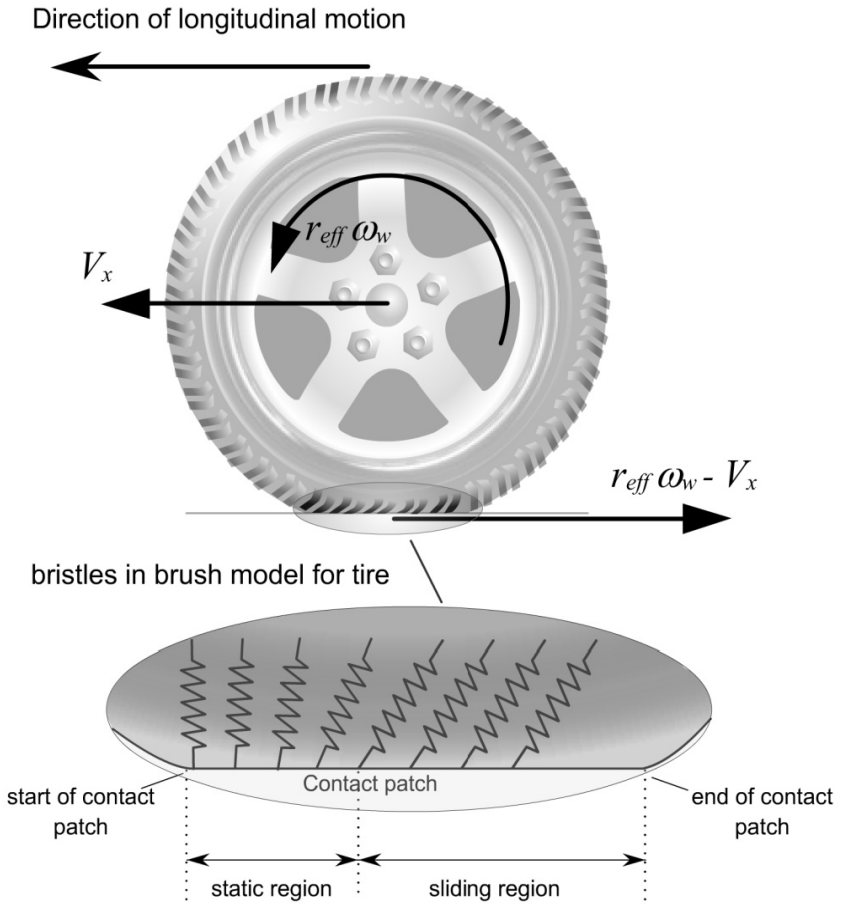


Figure 2.3: Longitudinal force in driving wheel [94].

Chapter 3

Learning-based Decision-Making at Urban Unsignalized Intersections: A Survey

Autonomous driving at unsignalized intersections is still considered a challenging application for machine learning due to the complications associated with handling complex multi-agent scenarios with a high degree of uncertainty. Automating the decision-making process at these safety-critical environments involves comprehending multiple levels of abstractions associated with learning robust driving behaviors to enable the vehicle to navigate efficiently. In this Chapter, we aim at exploring the state-of-the-art learning-based techniques implemented for decision-making applications, with a focus on algorithms that combine Reinforcement Learning (RL) and deep learning for learning traversing policies at unsignalized intersections. The reviewed schemes vary in the proposed driving scenario, in the assumptions made for the used intersection model, in the tackled challenges, and in the learning algorithms that are used. We have presented comparisons for these techniques to highlight their limitations and strengths. Based on our in-depth investigation, it can be discerned that a robust decision-making scheme for navigating real-world unsignalized intersection does not exist yet. Along with our analysis and discussion, we recommend potential research directions encouraging the interested players to tackle the highlighted challenges. By following our recommendations, non-overcautious, yet safe, motion planning models can be trained and validated in real-world urban environments.

In this Chapter, we illustrate the challenges and solutions for navigating unsignalized intersections in section **??**. We mainly focus on the start-of-the-art learning-based schemes

developed for tackling the intention inference challenge at unsignalized intersections. Then, with an emphasis on DRL-based schemes, we evaluate the learning-based decision-making schemes, emphasising our views on their intrinsic logic. Section 3.2 presents the possible research directions. Finally, Section 3.3 concludes the proposed and future works.

3.1 Unsignalized Intersection-Traversal: Challenges and Solutions

To enhance the AVs' ability to navigate complex urban unsignalized intersections, major navigational challenges need to be investigated. In this section, we survey these challenges which need to be taken into account while designing an automated learning-based decision-making algorithm for safe maneuverability at these safety-critical driving environments.

3.1.1 Autonomous Driving Under Uncertainty

The uncertainty associated with motion prediction of other intersections vehicles at unsignalized intersection is caused by the following factors [28]:

- **Unknown intention of intersection users**. The motion of other intersection participants is highly connected to the future trajectory of the ego vehicle [95]. Hence, for safe intersection navigation, precise motion predictions of the intersection users must be obtained. The main difficulty with inferring intention arises from the intrinsic uncertainty in the unknown current states and hidden variables, namely, unknown final destinations as well as their unforeseeable future longitudinal path [96], and their likelihood of interaction with the subject vehicle [**?**].
- **Noise characteristics of sensors' observations**. The noise associated with the measurements collected from the mounted sensors adds another layer of uncertainty to the decision-making problem.
- **Occluded environments and limited perception**. The ability to observe the scene accurately is hindered by environmental obstructions and occlusions. [97].

Fig. 3.1 depicts an illustrative example of where these uncertainties originate from at a four-way stop unsignalized intersection.

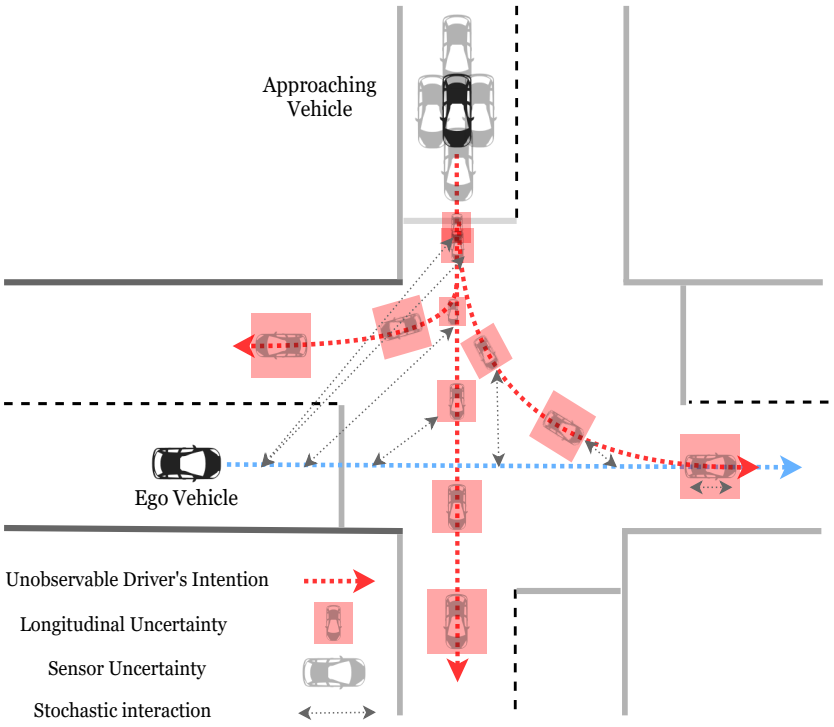


Figure 3.1: An intersection-traversal scenario where the ego vehicle is required to handle several sorts of uncertainties associated with the approaching vehicle.

Considering these uncertainties when designing learning-based decision-making schemes in a complex intersection environment is essential for the ego vehicle to traverse intersections safely. For example, predicted motion and future trajectories of the target vehicles [98], which share potential conflict points with the ego vehicle need to be incorporated while solving for an optimal traversal policy of the ego vehicle. This policy needs to be optimized for the most probable future scenarios coming from stochastic and interactive motion models of the other target vehicles. Considering these scenarios, in real time settings, these policies allow the autonomous vehicle to incorporate the estimated change in future prediction accuracy in the optimal policy [99]. This yields a compact representation with reduced-dimensions state-space

Based on our observation, we found that researchers have been mainly focusing at developing learning-based frameworks to tackle two main technical problems; inferring the intention of the intersection users and designing the decision process. Hence, in sections 3.1.2 and 3.1.3, we focus on exploring the published works on the unsignalized intersection-

traversal problem.

3.1.2 Driver Intention Inference Challenge

Accurately inferring and forecasting the intentions of drivers at unsignalized intersections is crucial for addressing the cause of an accident and ensuring road safety in such diverse multi-agent environments. Several research efforts have been exerted in order to develop algorithms for DII applications. These algorithms tackle the intention inference problem as a classification problem where intentions are classified based on the driving behaviour [100, 101]. These DII approaches can be classified into two groups: index-based and ML-based. In index-based approaches, safety metrics are utilized to examine driving behaviors at intersections in order to formulate risk assessment schemes. For example, time-to-intersection (TTI), time-to-stop (TTS), time-to-collision, Perception Reaction Time (PRT), Required Deceleration Parameter (RDP), along with brake application were taken into account for inferring the driver's intent at intersections [102, 103]. These index-based approaches, however, are designed for only frontal-crash prevention systems, where, in real driving scenarios, careless drivers may collide with the ego vehicle from different angles. ML-based classification techniques have been also employed for intention inference applications. For instance, Aoude *et al.* [80] proposed a Support Vector Machine-based (SVM) intention predictor that was developed as part of the proposed threat assessor scheme. Subsequently, the developed threat assessor warns the host vehicle with the identified threat level and advises the best escape path. Hidden Markov Models (HMM) were implemented for intention inference along with Gaussian Processes which were used for collision risks prediction of multiple dynamic agents [104]. Lefevre *et al.* [105] reported using a Dynamic Bayesian Network (DBN) for developing a probabilistic motion model where intentions are estimated from the joint motion of the vehicles. However, these ML-approaches fall short as they cannot capture the long-term temporal dependencies in the data.

Motivated by their efficacy in modelling sequential tasks, researchers have employed deep-structured Recurrent Neural Nets (RNN) for determining the intentions of drivers at non-signalized intersection. Zyner *et al.* [106] introduced the use of long short-term memory (LSTM) for intention inference at unsignalized intersections. Observations on the dynamic states, namely, position, velocity and heading states, were captured by the on-board set of sensors and used to train the network. In [107], a group of 104 features were utilized from the NGSIM dataset to train the proposed LSTM-based intention classifier. These features encompass ego position and dynamics, surrounding vehicles and their past states, and rule features which highlight what legal actions can be taken in the current lane.

The proposed method demonstrated high classification accuracy for intention prediction at intersections with different lanes or shapes. However, these methods rely heavily on the mounted on-board positioning/tracking system. This means that tracking data from GPS and Inertial Measurement Units (IMUs) are required in order for the system to operate effectively, restricting their usage to vehicles where streaming from these sensors is available. Zyner *et al.* [108] proposed a solution to this problem by using data from a Lidar-based tracking system which will be implemented in future intelligent vehicles. The proposed model was validated using a large naturalistic dataset which was collected from two days of driving at an unsignalized roundabout intersection. Recently, Jeong *et al.* [109] proposed a LSTM-based architecture for predicting the target vehicle's intention based on their estimated future trajectory at unsignalized intersections. This network was developed to study long-term dependencies between vehicles in complex multi-lane turn intersections, and was based on the previous sequential motion of the target vehicles measured by the sensors equipped with the AV. The predicted target motion is integrated with Model Predictive Control (MPC) which is responsible for planning the motion of the subject vehicle. Girma *et al.* [110] introduced the use of Bidirectional LSTM with an attention mechanism for intention inference at signalized intersections based on sequence-to-sequence modeling principles (i.e. Surrounding vehicles trajectory analysis with recurrent neural networks). Bidirectional LSTM is used due to its capability for exploring information from previous and future time steps. However, the proposed method is agnostic to the decision-making problem. Thus, integrating the proposed method with decision-making scheme in real-time format is a research direction to be explored. Table 3.1 summarizes the surveyed deep-learning-based intention inference schemes highlighting their research objectives and significant remarks.

3.1.3 Decision Making Challenge

Owing to the strengths of deep-structured neural networks in handling large partially-observable state-action space, major research directions have been followed aiming to develop learning-based schemes for tackling problems related to traversing unsignalized intersections autonomously. In this section, we present the main design challenges involved in developing learning-based algorithms for decision-making under uncertainty, as well as a review of relevant state-of-the-art solutions, emphasising key observations and shortcomings.

Partial Observability

In real multi-agent autonomous driving settings, the agents have incomplete information about the environment with which they interact. Therefore, designing a robust decision-making framework in such environments is considered an intractable problem. In practice, such problems are typically modelled as (POMDPs), in which a driving policy is learned to provide safe actions while accounting for the stochasticity inherent in the process of inferring intention and motion planning [116]. Numerous works address the problem of modeling the decision-making process of the partially-observable driving environments at unsignalized intersections. Brechtel *et al.* [117] models the decision-making problem for navigating an occluded T-junction intersection as a POMDP. Uncertainties of the driver's behavior and the limitations of the perception of the environment were taken into consideration while solving the continuous POMDP. Sezer *et al.* [118] develops a mixed observable MDP (MOMDP) model, which is a variant of POMDP, for intention-aware motion planning at a T-junction intersection under the uncertainty of drivers intentions. Along with the unknown intentions of other drivers, their unknown future predictions in the longitudinal direction and their interaction with the ego vehicle are modeled in the proposed decision scheme in [28]. The problem is formulated as a POMDP where the solution of the POMDP is a policy determining the optimal acceleration of the ego vehicle. However, the scalability of the proposed scheme to deal with unknown intentions of oncoming vehicles from multi-directions has not been addressed.

Inspired by the strengths of Deep Reinforcement Learning (DRL) approaches in learning driving policy without the necessity to learn the MDP model itself, several works have recently adopted these methods to solve the designed MDPs. For instance, Isle *at el.* [119] proposed a safe reinforcement learning algorithm for left turn intersection-traversal using action prediction techniques. An optimal policy is trained using deep Q-learning to minimize disruption to traffic which is measured by traffic braking and maximize the distance to other vehicles. To solve for an optimal policy in such a multi-agent environment, the problem was formulated as a Stochastic Game. Deep Q-learning Networks (DQN) have been used for solving intersection crossing problems modeled by POMDPs [34]. A thresholded lexicographic Q-learning scheme was adapted to the deep learning framework. This algorithm mimics human driving in some challenging scenarios where safety is prioritized over traffic rules and ride comfort. A factored MDP model was utilized instead of full MDP to mimic the human driver behaviour and to improve the data efficiency. A SUMO (Simulation of Urban Mobility) traffic simulator was then used as the simulation environment to validate the proposed experiment. Given the limitation of the Deep Q-Network, Bouton *et al.* [35] introduced an integration of the POMDP planning, model-checking and rein-

forcement learning to derive safe policies which can guarantee that the vehicle can traverse urban intersections under multiple occlusions and perception faults. Empirically, an ablation study was conducted showing that the proposed approach exhibits superiority over conventional DQN methods. A Deep Distributional Q-learning algorithm was proposed to deal with uncertainties associated with the variety of human driving styles [120]. The algorithm generates risk-sensitive actions based on offline distribution learning and online risk assessment. During the offline distribution learning, the distributions of the risk-neutral and state–action return are learnt from unknown behavior type of a participant sampled from a known environment. While the learned behaviour is being executed, the action risks (collisions) are quantified using distortion risk metrics where the optimal action can be then selected. Hoel *et al.* [121] introduced a method to evaluate the uncertain actions (decisions) made by the agent in an unsignalized intersection environment. A Bayesian reinforcement learning method using an ensemble of neural nets with Randomized prior Functions (RPF) [122], has been introduced to estimate the distribution of Q-values which are then utilized to estimate the action values. This proposed scheme shows robustness in identifying highly uncertain actions within and outside the training set which helps in choosing the safest actions for safe intersection traversing maneuvers. However, these proposed approaches fall short in terms of the proposed hard assumptions and the tailored intersection-traversal scenarios.

The development of robust DRL algorithms for better handling of POMDP problems has piqued the interest of many researchers in the field. Zhu *et al.* [123] introduced a scheme called Action-specific Deep Recurrent Q-Network (ADRQN) to improve the learning capability in partially-observable environments. A fully connected (FC) layer is utilized to encode actions which are coupled with their corresponding observations to form action-observation pairs. LSTM is then adopted to process the time series of action-observation pairs. Similar to the conventional DQNs settings, the FC layer calculates the Q-Values based on the latent states learnt by the LSTM network. Another LSTM-based Deep Recurrent Policy Inference Q-networks (DRPIQN) was also introduced to handle partial observability caused by imperfect and noisy state information in real-world settings [124]. Both ADRQN and DRPIQN networks outperform other deep Q-learning techniques in terms of learning capabilities and stability when applied to number of games. As an application to unsignalized intersection, Qiao *et al.* [33] proposed a network based on the design concepts of ADRQN and other deterministic gradient policy approaches for generating continuous time actions from the previous observations of the earlier steps. Observations from the previous 20 steps were used as inputs for the LSTM Network. Figure 5.1 exhibits the developed LSTM-Network which handles the POMDP and represents the decision-making problem of a four-way stop unsignalized intersection. The action output for each time step

is obtained based on the observation inputs to the first LSTM and FC layers of the network at each individual time step. Subsequently, the Q-values are generated by taking the action of the previous step a_{t-1} along with observation of the current step O_t as an input to the second LSTM and FC layers. However, these approaches are entirely model-free as they rely heavily on the LSTM network to remember the past instead of having true belief states. Igl *et al.* [60] proposed a Deep Variational Reinforcement Learning approach (DVRL), which relies less on a black box than the aforementioned DRPIQN and ADRQN, for learning optimal policies for POMDPs. Applying DVRL concepts for learning driving policy in partially-observable unsignalized intersection environments is still an area of research to be explored.

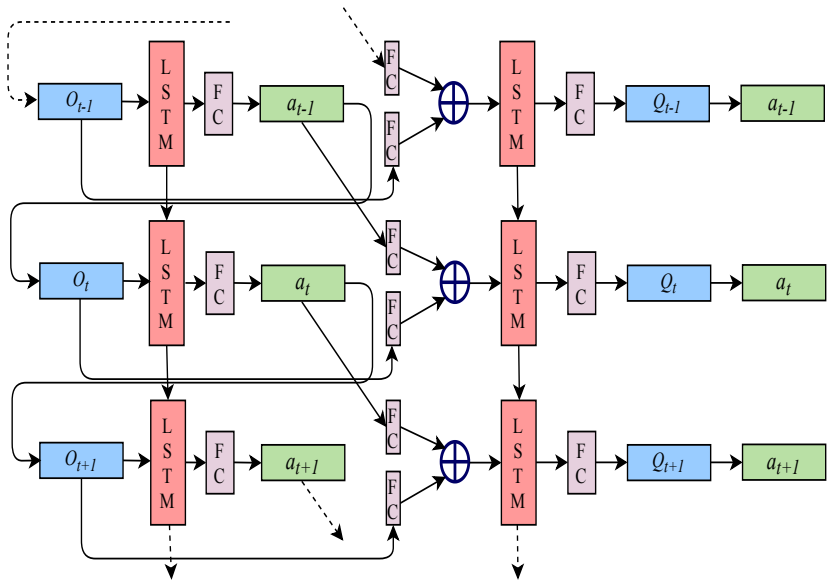


Figure 3.2: LSTM for solving the formulated POMDP of intersection-traversal problem.

Intention-aware schemes. These probabilistic decision-making algorithms were developed to control the motion under the unknown intentions of intersections participants. For instance, a continuous Hidden Markov model (HMM) was developed to infer the high-level motion intentions including turning and continuing straight [125]. A POMDP was then designed for the general decision-making framework, with assumptions and approximations used to solve the POMDP by calculating a policy to perform the optimal actions. Online solvers have also been used to solve the formulated POMDPS of the decision-making. In [58], an improved variant of the POMCP solver which is called The Adaptive Belief Tree (ABT) is used to solve the proposed POMDP of an intention-aware left turn-

ing problem. The proposed decision-making problem is based on mimicking the human behavior of creeping slowly, upon reaching the stop line, to better understand the driver's intention. The left-turn trajectory is simply assumed as a straight line with quarter circle curve. However, the intentions of the oncoming vehicles from only one direction is taken into consideration. In [31], the uncertainties associated with human behavior of other drivers on the road in the context of an intersection have been modeled as a POMDP. An online solver has been utilized to find the optimal action that can be taken by the vehicle to react to uncertain situations [31]. However, aside from a lack of real-world experimentation, using online solvers to solve is impractical because they only work for relatively small state spaces, and the complexity of solving the POMDP scales fairly quickly. Deep Reinforcement Learning, on the other hand, can work with much larger, or even large and continuous spaces, such as Atari [60]

Occlusion-aware schemes. As previously mentioned in 3.1.1, due to environmental uncertainties and the limited capabilities of the sensors on the autonomous vehicle, occlusions can pose significant challenges to safely traverse an urban unsignalized intersection. Hence, many research papers have addressed this problem while integrating risk assessors into the decision-making schemes. For example, an occlusion-aware algorithm for left turn maneuver risk assessment at four-way unsignalized intersections was developed [126]. A particle filter paradigm was utilized to represent the distribution of the possible unobserved potential locations (particles) of the vehicle. However, this algorithm is not representative of how the vehicle can make decisions, but can be coupled with any POMDP or any other decision-making algorithm. The same group, based on the forward and background reachability, developed a probabilistic risk assessment and planning algorithm for a four-way intersection. The algorithm borders the risk-inducting regions arising from the occlusions of the ego-perception sensors that can be used to generate collision-free routes [127]. However, none of these algorithms were tested in real-world environments. McGill *et al.* [32] addressed the problem of navigating unsignalized intersections in the presence of occlusions and faulty perception [32]. A risk model was proposed to assess the unsafe (risky) left turn across traffic at an intersection. Their model accounts for the traffic density, sensor noise and physical occlusions that hinder the view of other vehicles. By representing the intersection as a junction node with lanes entering and exiting the node, the risk assessment is used to determine a 'go' and 'no-go' decision at an intersection. The risk is modeled by defining near-miss braking incidents, collision incidents, traffic conflicts and small gap spacing. The risk is defined as the expected number of incidents that will occur if the ego-car enters an intersection. The overall risk is the sum of all expected incidents in all lanes and for all segments. In [37], the occluded intersection-traversal problem was viewed as a reinforcement learning problem. A deep Q-learning approach was utilized to traverse

a partially observed four-way intersection. A creeping behavior upon reaching the intersection is learnt where the agent must perform an exploratory action to better comprehend the environment. Three action representations were studied: Time-to-Go, Sequential Actions and Creep-and-Go. In the Time-to-Go representation, the desired path is known for the agent, and the agent decides whether to go or to wait at every point in time. While in the latter scenarios, the agent can determine whether to accelerate or decelerate progressively. However, a bird's eye view image space is used to describe the position of the vehicles in the space using Cartesian coordinates. This makes the implementation of the proposed DQN method inefficient for urban driving environment which require continuous actions rather than discrete ones. Moreover, in real AVs, it is infeasible to have a "bird's eye view"' for acquiring the vehicle's state for decision-making applications.

Table 3.2 summarizes the major classes of decision-making schemes under partial-observability.

Continuous action Space

In real autonomous driving, a continuous action of the autonomous agent is required for safe and efficient navigational tasks. DQNs, which are mostly adopted in the reviewed decision-making schemes, are used to learn an optimal policy for safety-oriented decision-making in a discrete action space domain. However, adapting such schemes to continuous domains, i.e. autonomous driving, is considered challenging, and in some instances, sample inefficient. Practically speaking, DQNs determine an action that has the highest action-value through an iterative optimization process at every step in the continuous action. For complex multi-agent decision-making including urban intersections, we have high-dimensional continuous action spaces. Discretizing these spaces to use conventional DQN schemes is not always an effective idea due to the exponential number of action values which might lead to the *Curse of Dimensionality*. Hence, to ensure convergence of the used model and capability, these continuous spaces must be handled in a robust way. Deep Deterministic Policy Gradient (DDPG) was adopted in [33, 132] for generating continuous actions rather than discrete actions for driving in four-way unsignalized intersection settings. Xiong *et al.* [133] presented an integration between Deep Reinforcement Learning and safety-based continuous control for learning optimal policy for self-driving and collision avoidance applications. DDPG, which adopts the actor-critic concepts (see Fig.3.3), is implemented to output the steering commands along with an Artificial Potential Field (APF) method for

collision avoidance and path planning applications. As this integration proves its usefulness for learning collision-free driving policies at highways, integrating such high-level DRL schemes with control laws can be vital for solving continuous control problems within the framework of unsignalized intersections. Recently, soft-actor-critic (SAC) algorithm has shown better performance in learning policies in continuous domains and stability characteristics than other deep deterministic algorithms including DDPG [71]. For autonomous driving applications, SAC has demonstrated remarkable ability in learning optimal policies for overtaking and maneuvering at roundabouts [56, 72]. Hence, applying soft-actor-critic (SAC) principles for learning traversing policies in complex driving scenarios can be a significant research avenue to be pursued.

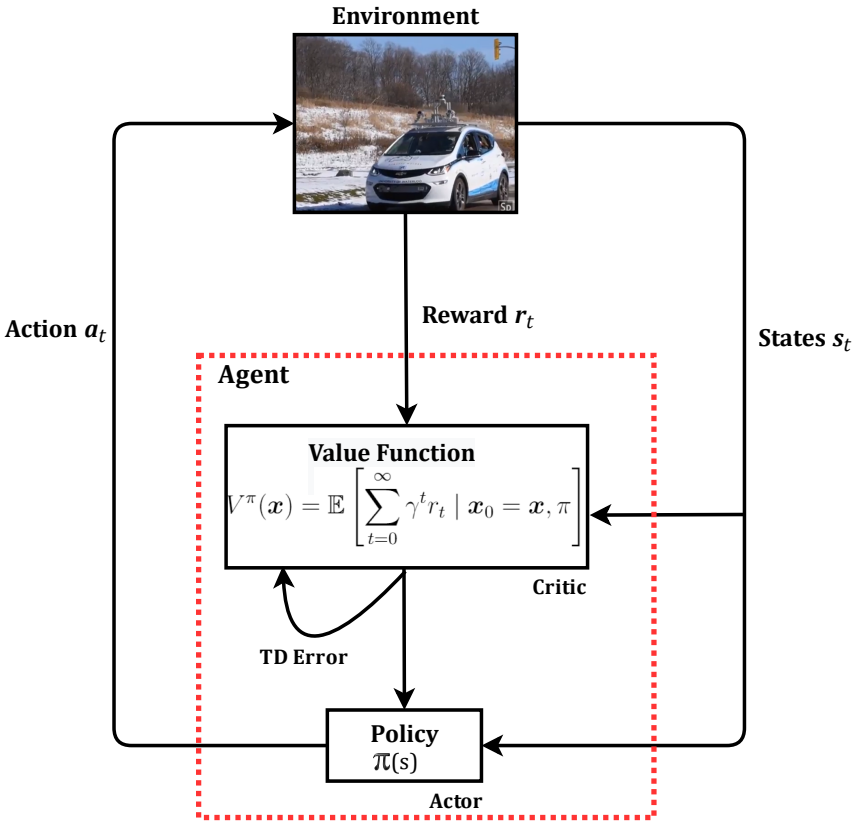


Figure 3.3: An illustrative sketch of Actor-Critic approaches. Actor-critic algorithms implement both the value-based approaches and the policy-based approaches. It comprises a couple of estimators: the actor network estimator which is based on Q-value, whereas the critic network utilizes the state-value function estimation.

Training in high-dimensional state-action space

As mentioned earlier in 2.3.1, DRL is centered on performing iterative optimization processes to learn a policy for a specific task. However, the number of iterations grows exponentially as the state-action space becomes larger. One discernible shortcoming of adopting DQN and DDPG is that an extensive training has to be performed in order to achieve an optimal behaviour. To accelerate the training process, Curriculum Learning (CL) approaches can be employed [55]. Qiao *et al.* [30] utilizes Curriculum Learning for reducing the training time and improving the performance of the agent in unsignalized intersection approaching and one-dimensional crossing behavior. However, applying CL concepts for other more complex scenarios. i.e. two-dimensional left-turn was not investigated. In [33], the same group proposed a DRL learning algorithm to traverse a four-way intersection with a two-way stop sign by taking into account the uncertainties that exist in the urban environment. This DRL algorithm is developed to utilize the preserved state-action values and the current LIDAR information along with the ego car's states information for designing the decision process. For efficient training in a high-dimensional space, a combination of LSTM and FC neural networks were designed to store the state-action pairs and generate continuous actions. Bouton *et al.* proposed a DRL algorithm for navigating urban intersections using the scene decomposition method [35] to improve training and to scale to a large number of agents. The decision-making under faulty perception is modelled as a POMDP. An extra state variable has been integrated to the global state vector to model the potential incoming traffic participant which is not present in the scene. A probabilistic model checker was adopted to compute the probability of reaching the goal safely for each state-action pair prior to learning a policy. Subsequently, a belief updater algorithm was developed to update the states uncertainties. Given the prior belief value and the current observations, the algorithm can integrate the perception error to the planning theme. It uses an ensemble of 50 Recurrent Neural Networks (RNN) to store the observation history. The training process was done using a synthetic dataset generated from a simulation environment. These techniques, however, have not been evaluated in real-world driving scenarios, where convergence of the proposed models is not guaranteed due to the breadth of possible crossing behaviors or directions of agents at unsignlized intersections.

Deep Reinforcement Learning from demonstrators has been introduced to enhance the learning capabilities, yielding a significant decrease in the total time of the training process by leveraging training sets with small demonstrations. Hester *et al.* [134] introduced Deep Q-learning from demonstrations (DQfD) to significantly accelerate the training process through leveraging sets with small demonstrations. A prioritised replay mechanism was

adopted for assessing the required data-sets ratio automatically. Nair *et al.* [135] proposed a technique based on DDPG and Hindsight Experience Replay for enhancing the training policies while learning the optimal policy for solving complex tasks using RL. Although, the proposed work has one major limitation which is the sample efficiency, a significant speed up of the training process was recorded. These works have led to other modifications of the training process of RL-based motion planning schemes. For instance, Hierarchical Reinforcement Learning (HRL) architecture was developed based on inclusion of heurisitic-based rules-enumeration policy to enhance agent's exploration for behavioral planning at intersections [136]. More recently, Huang *et al.* [137] developed an integration between the imitative expert priors from expert demonstrations for learning driving policies at urban environments. The priors of the imitative expert are learnt using imitation learning and uncertainty quantification method, which governs the learning performance of the agent while still allowing it to explore. In brief, Table 3.3 epitomizes the limitations of the most relevant research works on decision-making using reinforcement learning-based schemes.

3.2 Discussion and Research Directions

In our discussion provided in sections 3.1, we emphasised key points on how the examined systems might be further enhanced to better address the decision-making problem under uncertainty at unsignalized intersections. However, there are still significant gaps in our understanding of how to advance our efforts toward creating high-fidelity frameworks that can operate autonomous cars in real-life environments.

Our thorough analysis has revealed that there is currently no reliable decision-making method for negotiating real-world unsignalized intersections. We discern that the state-of-art decision-making schemes focused on the high-level decision making layers, i.e high-level reasoning for behavioral path planning, neglecting other low-level layers proposed earlier, including low-level motion planning and control. Furthermore, implementation and testing in real-world driving environments is not investigated. In practice, convergence of the RL-models in simulation-based environments does not necessarily ensure generalizability in real-life scenarios due to the domains mismatch. Real-world observations differ in terms of the associated noise sequences and vehicle dynamics response. We therefore suggest fundamental research directions, in this section, based on these observations in order to advance research in this area.

3.2.1 Low-level local planning and control integration

Numerous research papers have addressed the low-level motion planning problem and control at urban unsignalized intersections using Model Predictive Control (MPC) principles. For instance, Hu *et al.* [63] proposed an event-triggered model predictive adaptive dynamic programming technique for motion planning at urban intersections. The method takes urban speed, vehicle kinematics and road constraints into consideration while solving multi-objective optimization problem. However, for high-fidelity decision-making applications in urban autonomous driving, incorporating the local motion planning and low-level control layers taking into account vehicle dynamics is essential to ensure the feasibility of the high-level RL commanded actions. Such integration has been done for intersection-management applications, where centralized reference signals being distributed to the intersections agents via V2V communication [64, 65]. In [65], an integration between high-level decision-making layers and low-level MPC-based motion planing layer has been proposed for learning supervisory intersection-management policy in connected driving fashion. However, as per the authors' knowledge, such integration has not been developed for learning intersection-traversal policy of the learning agent. Hence, having the motion planing layer integrated while approximating intersection-traversal policies would facilitate learning, with feasibility guarantees [67], tacking into account lateral and longitudinal dynamics.

In connection with autonomous driving problem in adverse weather environments [140, 141], incorporating high-fidelity vehicle dynamic models is also critical for longitudinal and lateral motion planning. For instance, learning safety-oriented policies for intersection-approaching behavior at unsignalized intersection, where braking is applied to decelerate smoothly for precise stopping, is a prerequisite condition for safe intersection navigation. Hence, learning an optimal deceleration profile (curve) $a_{x,optimal}$, which ensures a comfortable ride while remaining efficient, requires an inclusion of longitudinal vehicle dynamic models and braking performance which is coupled with the road surface conditions represented by friction coefficients (see rough curves in Fig. 3.4).

3.2.2 Real experimental validation

As can be seen from Table, 3.3, most of the reviewed schemes have been tested in simulation-based environments. This can be valid, as RL techniques require collecting a large amount of real world based training data which would be costly in terms of effort and time. Practically speaking, simulated observations, which are streamed from modelled sensors, have

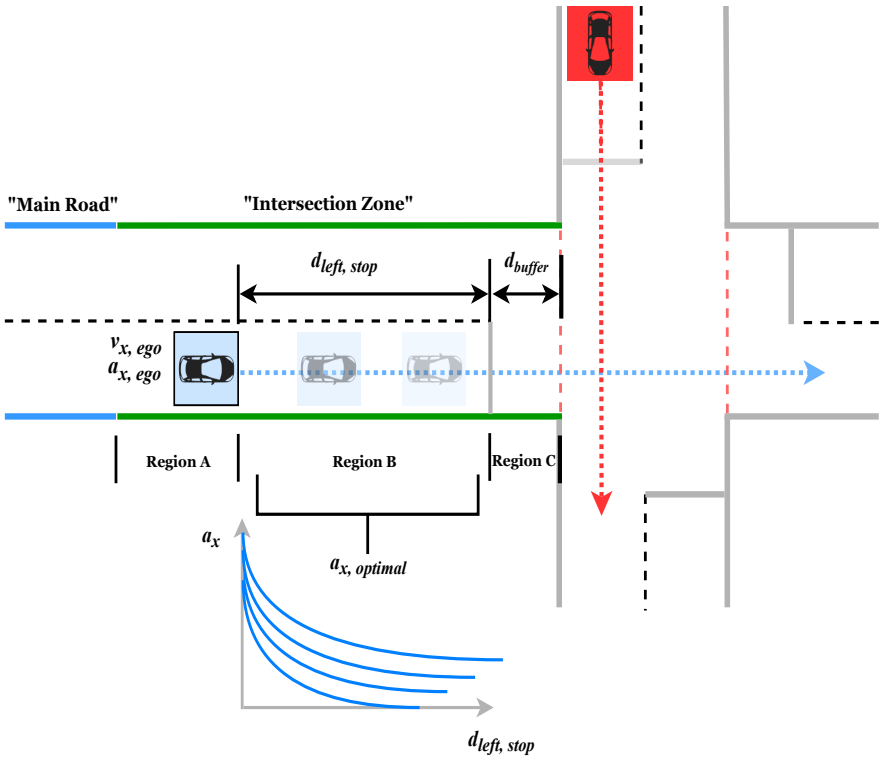


Figure 3.4: An illustrative sketch of the intersection-approach phase scheme. As shown, the vehicle enters **Region A** with the standard speed $V_{x,ego}$ of (40–50 km/h). In **Region B**, the vehicle is assumed to start decelerating with rate $a_{x,ego}$ to reach the stop-line. **Region C** represents the safety buffer d_{buffer}.

different data distributions compared to real data which may lead to failure in generalization on (unseen) real data [142]. Sim-to-real transfer learning techniques have been introduced to further promote training RL approaches in real environments [143].

Domain Adaptation (DA) and Domain Randomization (DR) techniques have also been proposed to enhance the generalization capabilities of ML-based models on a target domain. Feature-level DA methods are designed to learn domain-invariant features which cannot discriminate between the source and the target domains, whereas pixel-level DA techniques focuses on shaping images from the source domain to be analogous to the target domain's images using Generative Adversarial Networks (GANs) [144, 145]. Ganin *et al.* [146] describe a domain-adversarial training of neural networks for Feature-level domain adaptation. This model is based on features that are discriminative for the central learning process, but indiscriminative with the translation between these domains. In

[147], an end-to end (i.e., perception to control) transfer learning using image-to-image translation for domain transfer was applied for autonomous driving. Although the lane following driving policy was learnt from the simulation domain with control labels, the model was able to provide control from real images due to the shared latent space between the two domains. DR methods are based on the concept of exposing the learning agent to the stochastic random environment where its properties can be randomized in order to augment robustness in real-world deployment [148]. Recently, Amini *et al.* [149] introduced a training engine for transfer learning of end-to-end autonomous driving policies using sparsely-sampled trajectories from human drivers. Utilizing these trajectories has yielded robustness in performing real driving tests in unseen complex and near-crash environments. Using CARLA, the performance of the proposed method has been evaluated in comparison with the DA and DR approaches. The results exhibited superiority of the proposed approach over the conventional transfer learning approaches in terms of recovery from hazardous near-crash situation.

In short, validating the RL approaches in real-world driving settings is an active area of research. Inspired by the presented DR and DA techniques which prove its robustness in learning optimal policies for end-to-end autonomous driving, the real-world experimental validation of the simulation-based decision-making approaches would be further facilitated by creating real-life intersection driving scenarios.

3.3 Conclusion

Unsignalized intersections are safety-critical areas in urban environments due to the complex driving behavior and the lack of traffic control signals. Consequently, developing robust decision-making and motion-planning for these multi-agent environments is highly intractable due to the complexities associated with the partially observable multi-agent driving environment and the environmental uncertainties. With the resurgence of deep learning, modern RL techniques have been utilized to handle such problems with a large space of observations to learn safe driving policies.

This Chapter reviews various aspects related to challenges associated with decision-making at unsignalized intersections with a focus on learning-based schemes. We discuss these schemes in terms of the tackled driving scenario, the involved challenges, the proposed learning-based designs and the validation in simulations and real-world environments. Based on our discussion and investigation, we found that research efforts are still required to tackle the real-world challenges of unsignalized intersection-traversal problem.

Ultimately, the decision making schemes that were reviewed have been proposed to tackle uncertainties associated with traversing the unsignalized intersection problem. This is commonly modeled as a POMDP due to the unknown intention and future trajectories of intersection users. Environmental uncertainties due to limited sensor range and faulty perception are taken into account while designing occlusion-aware decision making schemes. Furthermore, uncertainties of different driving styles are also considered in developing decision-making schemes for learning optimal crossing policy in multi-agent environments. However, we noticed critical areas have received little attention and lack in-depth research. These areas are related to the lack of utilization of high-fidelity vehicle dynamic models and the experimental validation of the proposed decision making schemes. However, if robust motion planning is designed and integrated to account for these critical environments, a high-fidelity vehicle dynamic model must be used to reflect the vehicle response precisely. On the other hand, we suggest methods and heuristics that can be used to facilitate real-world driving for testing and validation purposes of the RL-based models. Taking our recommendations into consideration, precise and non-overcautious motion planning models can be trained and validated in real-world urban settings.

Table 3.1: Summary of the covered Deep-learning-based intention inference schemes in this section

Ref.	Objective	Method	Remarks
[106]	Intention inference based on the ego vehicle's observations i.e. GPS, IMU and Odometry).	RNN	• 100 % classification accuracy on the Naturalistic Intersection Driving Dataset [111].
[107]	Intention inference at multi-lane intersection based on observations of speed, lanes and six adjacent vehicles.	LSTM	• 85% accuracy at intersections with different types and shapes (NGSIM) dataset.
[108]	Intention inference for ego vehicles without tracking data (GPS, steering wheel encoding).	LSTM.	• The results indicate that networks fed with more history up to 0.6 seconds performs better. • The provided model gives 1.3 sec prediction window prior to any potential conflict.
[109]	Intention inference based on GPS, Lidars and different types of cameras (front and round views).	LSTM	• Based on the prediction results, longitudinal motion planning with safety guarantees is proposed using MPC .
[110]	Intention Inference based on focusing on important time-series vehicular data.	Bidirectional LSTM with attention mechanism.	• Sequence to sequence modeling is performed to map the input sequence of observation to a sequence of predicted driver's intentions. • Achieved high accuracy on the NDS dataset. [112]
[113]	Intention inference for maneuver prediction at intersection based real driving sequences including vehicle dynamics, gaze data as well as head movements.	LSTM	• A prediction window of 3.6s has been achieved on RoadLab dataset [114].
[115]	Intention inference for Path prediction using dilated convolution networks in conjunction with a mixture density network (MDN) considering the temporal aspects of driving data.	Temporal CNN	• Outperforms ML-LSTM and LSTM-FL in terms of accuracy and computational complexity on ACFR dataset.

Table 3.2: Classes of decision-making schemes under under partial-observability at unsignalized intersection

Class	Contribution	References
Occlusion-aware	• Navigation through static and dynamic occlusions.	[128] [30] [117] [129]
	• Navigating under perception errors due to occlusions	[35] [32] [119]
	• Navigating with Limited sensor range	[130] [33] [126]
	• A creeping behavior is learnt to better comprehend the environment.	[37]
Intention-aware	• SVM-based motion planning.	[131]
	• Target motion-based behavioral planning.	[29]
	• Navigation through unknown intention and noisy perception.	[28]
	• Inferring High-level motion intentions including turning and going straight.	[125, 57]

Table 3.3: Overview of the reviewed Reinforcement learning-based decision-making schemes at unisgnalized intersection

Ref.	Intersection Type	Method	Data Collected	Remarks	Limitations
[30]	4-way Unsignalized (stop-sign)	DRL using automatic generation of curriculum for training enhancement	Simulation-based	• A more realistic 4-way intersection driving scenario is proposed where vehicles are programmed not to yield to ego vehicle if it is in their FOV	• Environmental uncertainties, which cause errors in perception were not considered while collecting Observations from simulated sensor (LIDAR + Cameras)
[37]	T-junction	DQN	SUMO simulator [138]	• A creeping behavior upon reaching the intersection is learnt where the agent must perform an exploratory action to better comprehend the environment	• A god-view state space is used to describe the motion of vehicles at intersection (Not true for real-life driving)
[34]	Multi-lane 4-way intersections and round-abouts.	Multi-objective RL (Thresholded lexicographic Q-learning)	Collected Via SUMO simulator	• Learning safe crossing with the presence of faulty perception and occlusion • The trained policy is scalable across a range of urban roads with different shapes • Learning human behavior of looking at vehicles at area of interest	• A full knowledge of other vehicles based on a bird's eye view representation of state space (not realistic for real-world) • Not tested in real-world environments
[33]	4-way Unsignalized (stop-sign)	RL-based approach using hierarchical option	Collected Via SUMO simulator	• Learning an optimal policy for robust traversing under environmental uncertainties • Results shows superiority over the rule-based techniques and classical approaches	• No guarantees for possible scalability at more complex intersections with multi-lanes • Not tested in real-world environments
[35]	T-junction	Integration of model-checker and RL	Simulation-based	• Learning safe crossing with the presence of faulty perception and occlusion	• The proposed method was not validated through real testing to show the validity of the simulated POMDP-based simulated values of the perception errors
[136]	Multi lane 4-way intersection	Hierarchical reinforcement learning with hybrid reward mechanism	MSC's VIRES VTD (Virtual Test Drive) simulator	• Better convergence capabilities and sample-efficiency compared to the classical RL Methods	• Focus on mimicking human driving in limited go-straight and left-turn maneuvers • Not tested in real-world environments
[129]	Multi-lane 4-way intersection.	DQN	Collected Via CARLA Simulator [139])	• DQN shows less overcautious behavior under limited sensor range and faulty perception compared to the rule-based algorithms	• DQN is utilized for learning the driving policy. However, DQN is restricted to the discrete action domain • Not tested in real-life environments
[61]	4-way	DQL and DDQL	Simulation-based	• The proposed results show safe and repeatable Left-turn maneuver is learnt where the collision rate is significantly reduced	• Training based on simulated sensor observations. • The proposed scheme is restricted to discrete action space.
[121]	4-way	Bayesian RL-based scheme using an ensemble of NN with Randomized Prior Functions (RPF)	Simulation-based	• The Uncertainty of the RL agent's actions is estimated.	• lacks real-world testing • Assumptions related to the formulated decision-making problem have been made, i.e. the environment is assumed to be fully observable (MDP)
[36]	T-junction	RL with stochastic guarantees	Simulation-based	• Traversing with safety guarantees	• The proposed scheme deals with discrete action space only • Assumptions made for the vehicle and the pedestrians motion
[62]	4-way	DQL and DDQ	Simulation-based	• RL-enabled control framework is built using transfer rules	• RL scheme deals with discrete action space only. • The proposed geometric controller does not represent actual vehicle constraints, e.g. max steering rate.

Chapter 4

Cyber-Physical System State Estimation in Urban Driving

Improving the performance of the Safety-Critical Cyber-Physical (CPS) system in today's new electric vehicles is critical for the vehicle's safe manoeuvrability. The braking system, as a typical CPS system, is critical for vehicle design and safe control. As a typical CPS system, the braking system is crucial for the vehicle design and safe control. However, precise state estimation of the brake pressure is desired to perform safe driving with a high degree of autonomy. In this Chapter, a novel sensor-independent state estimation framework for a safety-critical state estimation technique of the vehicle's brake pressure is developed using a deep-learning approach. A (DNN) is structured and trained using special deep-learning training techniques, such as, dropout and rectified units to obtain more accurate neural inference model for brake pressure using indirect measurements of vehicle dynamic states and powertrain states. The proposed model is trained using real experimental training data which were collected via conducting real vehicle testing. The vehicle was attached to a chassis dynamometer while the brake pressure data were collected under random driving cycles. Based on these experimental data, the DNN is trained and the performance of the proposed state estimation approach is validated accordingly. The results demonstrate high-accuracy brake pressure state estimation with RMSE of 0.048 MPa.

This Chapter proposes a sensor-independent deep-learning-based approach for precise state estimation of the electric vehicles brake pressure. The phrase "*sensor-independent*" refers to that there is no sensor required to capture the changes in the braking pressure cycle while driving. Because of the enormous noise associated with the onboard vehicle's sensors, implementing sensor-independent data-driven estimate algorithms is critical for designing

robust control systems [51, 150]. After exploring the published schemes, utilizing modern DL techniques for state estimation of brake pressure has not been addressed.

The remaining sections of this Chapter are organized as follows: Section 4.1 reviews the state-of-the-art state estimation schemes of the vehicle's cyber-physical systems. Section 4.2 demonstrates the dropout-based training and ADAM optimization techniques. Section 4.3 illustrates the experimental setup and data collection system. Section 4.4 presents the proposed DL brake pressure estimation technique. In Section 4.5, the preliminary results of the proposed brake pressure state estimation approach. In Section 4.6, the proposed and the future works are presented.

4.1 Introduction

With the rise of autonomous vehicles, Cyber Physical Systems (CPSs) have become a major research focus, with teams from academia, industry and government organizations studying them [151, 152, 153, 154]. The various subsystems of the electric vehicle, like communications, electric powertrain and energy management, sensors, the driver, and the environment all come together to form a tightly coupled, dynamically interacting system [155, 156, 157] . The resulting system has strong uncertainties, nonlinearities, and difficult to model interactions between its parts, making estimation and control of CPSs in EVs a difficult task.

For CPSs in EVs, we are concerned primarily with safety critical systems, such as the braking system [46, 47, 48]. Braking systems have benefitted from numerous technological advances in the last several decades, such as new control schemes, higher safety standards, and other electronic improvements [158, 159, 160]. With increased autonomy and control authority, however, it becomes increasingly important that the braking system be accurate and safe against faults. Braking control generally uses measurements of the hydraulic pressure in brakes to decide actions to be taken, measured by pressure sensors [49]. If a hardware or software fault occurs in these sensors, however, brake control can be compromised, leading to potentially dangerous safety issues. This can be circumvented by using high precision brake pressure state estimation, with the potential to evolve the system into a sensor-independent system with sufficiently accurate estimation [50]. This type of estimation has been a hot research topic in the past, generally through the use of control theory-based approaches. A recursive least-squares approach was used in [161] to estimate brake pressure by using characteristics of the pressure response of antilock braking systems. An Extended Kalman Filter based approach, combining tire dynamics and hydraulic models, was used in [162]. Other approaches have included the design of

inverse models for brake pressure [163], modeling the decrease, increase, and hold of brake pressure using experimental data [164], including measuring the amount of fluid passing through the valve to determine brake pressure [165]. All of these, however, are controls based approaches, with none being suitable for a fully sensor-less design. [51] uses a neural net to perform estimation of brake pressure, using data obtained from an EV. However, authors used conventional back propagation while training the FFNN, which suffers from problems with overfitting, vanishing gradient, as well as higher computational complexity in training. Nevertheless, these problems have been resolved by implementing the recent advances of Deep Learning techniques to augment the training process of the DNN [52].

Deep learning can be described as a learning approach that employs DNNs which comprise of two or more hidden layers [166]. Deep learning was introduced to resolve the problems associated with the poor training techniques used with DNNs [167]. Among these problems are the overfitting and vanishing gradient problem. The vanishing gradient problem has been resolved using rectified unit functions as activation functions [168]. The overfitting problem has also been tackled by implementing modern regularization techniques, such as dropout [169]. Dropout randomly drops units and their connections during the weight update cycle to reduce overfitting. Similar results were found by [170], which used rectified linear units (ReLU) and dropout techniques to improve the accuracy of their neural net. These advantages are exploited to improve the training accuracy of the DNN and enhance the brake pressure state estimation.

Deep learning is also being used prominently in various forms in industrial applications [171]. Fault detection and classification is an active research topic, with deep learning being used both to identify faults and to classify them, as well as for learning features to be used in fault classification [172, 173]. LiftingNet uses a multilayer neural network to extract deep features from noisy data for use in fault classification in motor bearings [174]. Other approaches have also been used, such as one that used unsupervised learning vibration images to perform intelligent feature extraction for fault diagnosis in rotor systems [175]. Semi-supervised learning is a more common approach, with one approach using semi-supervised learning based on hierarchical extreme learning machines for soft sensor modelling [176]. Recently, Dendritic Neuron Model (DNM) with learning schemes has shown significant ability in solving classification and estimation problems [176].

4.2 DL-based State Estimator Design

Neural networks have demonstrated good performance for state estimation of the brake pressure [51]. However, obtaining highly expressive state estimator's model is linked, in

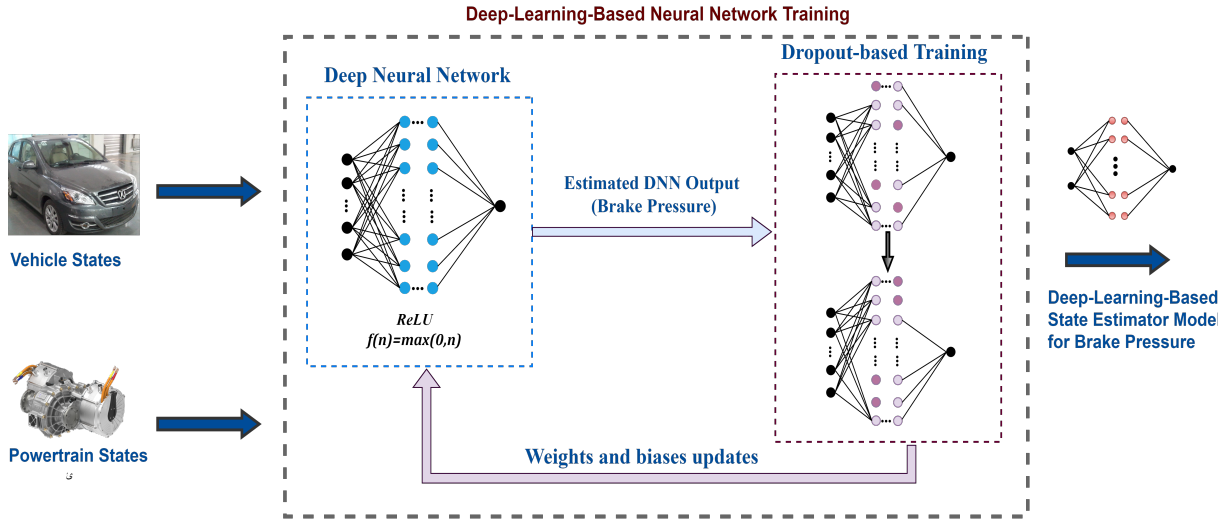


Figure 4.1: Proposed state estimation framework.

practice, to the accuracy of the utilized training method. Despite the effectiveness of conventional training techniques, modern deep-learning-based structures have shown superiority in terms of minimizing the associated overfitting and achieving fast convergence. In this section, a Deep Neural Network (DNN) is proposed using dropout regularization to improve the quality of the brake pressure estimation. As described in Fig. 4.1, the neural inference model uses indirect measurements including vehicle states and powertrain states as inputs and the ground truth of the brake pressure values are the outputs while conducting the training experiment of the neural network. In this section, the standard structure of the DNN is illustrated. Dropout and the ADAM optimization technique are also presented.

4.2.1 Dropout-based Training

The backpropagation algorithm is commonly used for updating the weights of the NN. The operation of a neural network can be described using the equation:

$$a^{m+1} = \boldsymbol{f}^{m+1}\left(\boldsymbol{W}^{m+1}a^{m} + \boldsymbol{b}^{m+1}\right) \tag{4.1}$$

where a^{m} and a^{m+1} represent the outputs of the mth and $m+1th$ layers of the FFNN, and $bm+1$ is the bias weights for the $m+1th$ layer. While training, the aim is to train the net-

work with associations between the specific input-output mappings $\{(p_l, t_l), \ldots (p_Q, t_Q)\}$, where p is the input vector and t is the associated output. The Backpropagation algorithm adopts the Mean Squared Error (MSE) as the performance index for optimization, which can be approximated as:

$$F(x) = e^T(k)e(k) \tag{4.2}$$

The steepest descent algorithm, using F as above, is then

$$\begin{aligned} w^m_{i,j}(k+1) &= w^m_{i,j}(k) - \alpha \frac{\partial F}{\partial w^m_{i,j}} \\ b^m_i(k+1) &= b^m_i(k) - \alpha \frac{\partial F}{\partial b^m_i} \end{aligned} \tag{4.3}$$

where a is the learning rate. Defining the sensitivity of F to changes in the ith element of the net input at layer m as

$$s^m_i = \frac{\partial F}{\partial n^m_i} \tag{4.4}$$

the derivatives in (6) and (7) can then be simplified to

$$\frac{\partial F}{\partial w^m_{i,j}} = s^m_i a^{m-1}_j \tag{4.5}$$

$$\frac{\partial F}{\partial b^m_i} = s^m_i \tag{4.6}$$

which then allows for the approximate steepest descent to be described in matrix form as a Jacobian with form

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial n^{m+1}_1}{\partial n^m_1} & \cdots & \frac{\partial n^{m+1}_1}{\partial n^m_{sm}} \\ \vdots & \ddots & \vdots \\ \partial n^{m+1}_s & \cdots & \frac{\partial n_s m+1}{\partial n^m_s m} \end{bmatrix} \tag{4.7}$$

where each element can be expressed as

$$\frac{\partial n^{m+1}}{\partial n^m} = \mathbf{W}^{m+1} \dot{\mathbf{F}}^{\mathrm{m}} (\boldsymbol{n}^m) \tag{4.8}$$

and

$$\dot{\mathbf{F}}^{\mathrm{m}}\left(\mathbf{n}^{m}\right)=\begin{bmatrix} \dot{f}^{\mathrm{m}}\left(n_{1}^{m}\right) & 0 & \cdots & 0 \\ 0 & \dot{f}^{\mathrm{m}}\left(n_{2}^{m}\right) & & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & f^{\mathrm{m}}\left(n_{s^{m}}^{m}\right) \end{bmatrix} \tag{4.9}$$

The recurrence relation for sm can then be expressed using the chain rule

$$\begin{aligned} s^{m}=\frac{\partial F}{\partial \mathbf{n}^{m}} &= \left(\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^{m}}\right)^{T}\frac{\partial F}{\partial \mathbf{n}^{m+1}} \\ &= \dot{\mathbf{F}}^{m}\left(\mathbf{n}^{m}\right)\left(\mathbf{W}^{m+1}\right)^{T}s^{m+1} \end{aligned} \tag{4.10}$$

This relation can then be initialized at the final layer as

$$s_{i}^{M}=\frac{\partial \boldsymbol{F}}{\partial n_{i}^{M}}=\frac{\partial\left((\mathbf{t}-\mathbf{a})^{T}(\mathbf{t}-\mathbf{a})\right)}{\partial n^{M}} \tag{4.11}$$

$$\begin{aligned} &= \frac{\partial \sum_{j=1}^{SM}\left(t_{j}-a_{j}\right)^{2}}{\partial n_{i}^{M}} \\ &= -2\left(t_{i}-a_{i}\right)\frac{\partial a_{i}}{\partial n_{i}^{M}} \\ &= -2\left(t_{i}-a_{i}\right)\dot{f}^{\mathrm{m}}\left(n_{i}^{m}\right) \end{aligned} \tag{4.12}$$

The final recurrence relation can thus be summarized as

$$s^{M}=-2\dot{\mathbf{F}}^{M}\left(\mathbf{n}^{M}\right)(\boldsymbol{t}-\mathbf{a}) \tag{4.13}$$

The training procedure of the neural network is deemed sensitive, and utilizing the conventional backpropagation (BP) as a stand-alone training approach may result in inaccurate performance of the obtained model [177]. BP experiences some problems pertaining to overfitting and computational complexity, which was recently tackled using some innovative DL training techniques, namely the use of dropout.

Dropout is a DL technique that was introduced as a simple way to resolve the problem of overfitting [52]. As shown in Fig. 4.2, Dropout considers randomly selected units for

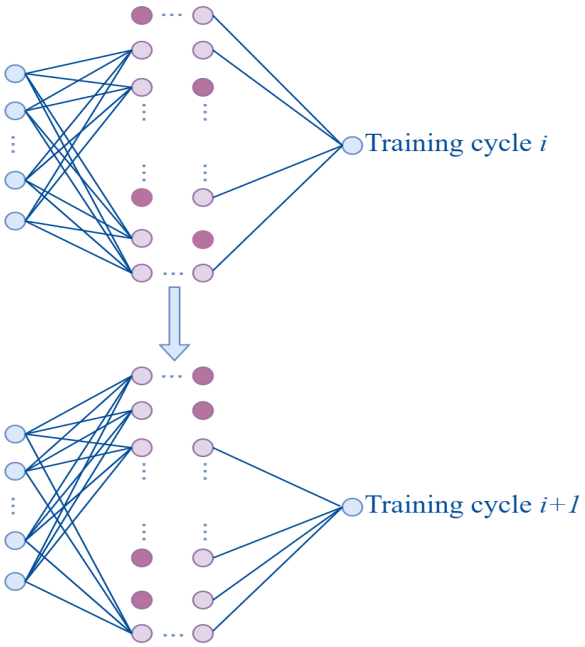


Figure 4.2: Dropout Training Technique

training rather than all units [170]. In each layer m, p^m denotes a vector of independent Bernoulli random variables which represents the probability of the dropped-out nodes. This vector is multiplied element-wise with the output of the associated layer, a^m, to form the thinned outputs, $\widehat{\mathrm{a}}^m$, as $p_j^m \sim$ Bernoulli (p)

$$\widehat{\mathbf{a}}^m = p^m * \mathbf{a}^m \tag{4.14}$$

The feed-forward operation, with dropout, is formulated as

$$\mathrm{a}^{m+1} = \mathbf{f}^{m+1}\left(\boldsymbol{W}^{m+1}\widehat{\mathrm{a}}^m + \boldsymbol{b}^{m+1}\right) \tag{4.15}$$

4.2.2 Training Optimization

Dropout has shown its usefulness for Bayesian approximation and estimation purposes [178]. Incorporating adaptive gradient-based optimization techniques, such as, Adaptive Moment Estimation Technique (ADAM) with dropout has shown an excellent regression performance with a minimum training cost [179]. Hence, in this study, ADAM is adopted

for gradient-based training optimization due to its fast convergence in regression problems compared to the conventional Stochastic Gradient Descent (SGD) [179].

Based on the estimation of the first and the second moment of the gradients, ADAM calculates the adaptive learning rates for specific parameters. The optimization algorithm has been considered suitable in applications where data contains a large number of parameters. ADAM optimization is also well-suited to state estimation problems where the measurements are associated with immense noise sequences and sparse gradients [179]. Fig. 4.3 represents the flowchart of the ADAM optimization scheme. α and t define the step size and time step; respectively. θ_0, m_0 and v_0 represent the initial parameter vector, first and second moment vectors.

The main aim of the ADAM optimization technique is to minimize the expected value of the noisy objective $f(\theta)$ with regard to its parameters θ. $f_1(\theta)_{,\ldots,f_N}(\theta)$ represent the outputs of the stochastic function of following timestamps $1, \ldots\ldots, N$. The sources of stochasticity might be associated with evaluation of the randomly selected batches of data points and/or from the associated noise sequence of the objective function. The vector of the partial derivatives of the objective function f_t can be described as the gradient with respect to θ.

$$g_t = \nabla_\theta f_t(\theta) \tag{4.16}$$

The squared gradient (θ_t) and the mean and the moving averages of (m_t) the mean represent the estimates of the variance and the mean of the gradient, respectively. The exponential decay of these averages are controlled by using hyper-parameters β_1 and β_2.

4.3 Experimental Setup and Data Collection

In this section, the experimental setup as well as the data collection procedure is presented. The proposed DNN is trained using real vehicle driving data. Several experiments were performed to collect training data using an electric passenger vehicle with a chassis dynamometer attached. The testing vehicle, data collection methods, selected feature methods, testing scenarios, and data pre-processing are described in the following sections.

4.3.1 Testing vehicle and the electric powertrain

The data were collected via conducting real driving experiment using an electric car with chassis dynamometer, as exhibited in 4.4. The chosen vehicle is driven by a permanent-

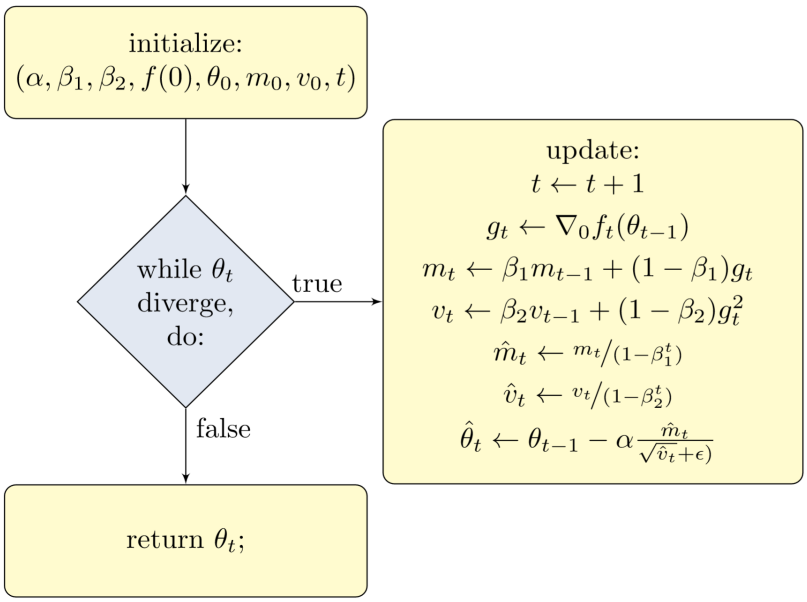


Figure 4.3: Adam optimization scheme

magnet synchronous motor, which can operate either in driving or generating modes. The electric motor is powered via battery via a DC bus, which releases or absorbs the electric power during driving or braking cycles, respectively. Relevant specifications of the test vehicle and power train are shown in Table 4.1.

Several driving cycles standards can be used to set up the testing scenarios. The New European Drive Cycle (NEDC) which comprises of four repeated urban driving cycles was used for brake pressure estimation using NN [51]. However, arbitrary driving style imposes more challenges for NN to predict the brake pressure. In this paper, random driving cycles were adopted to better represent the urban driving environment with a range of speed up to 45 km/h. The proposed DL based brake pressure estimation is designed to capture the relevant trends of the urban driving behavior.

The electric powertrain comprises the fundamental components that generate electrical power. This comprehends a gearbox, an electric motor, a differential and a couple of two half shafts. At the center of the front axle, the electric motor is placed. In acceleration scenarios, accelerating the vehicle, the electric motor produces a propulsion torque that is transmitted to the axle to propel the vehicle through the drivetrain. While the electric motor switches to the regenerative brake mode to apply a braking toque in the vehicle deceleration scenarios [180, 181].

Figure 4.4: EV testing using a chassis dynamometer.

Table 4.1: EV and Powertrain list of specifications.

	Specification	**Value**	**Unit**
Vehicle	Overall Mass	1360	kg.
	Gear ratio	7.881	-.
	Transmission efficiency	96%	-.
	Nominal radius of tire	0.295	m.
	Coefficient of air Resistance	0.32	-.
	Wheelbase	2.50	m.
Battery	Voltage	326	V
	Capacity	66	Ah
Electric motor	Maximum torque	144	Nm
	Peak power	45	kW

The hydraulic brake system installed in the testing vehicle includes wheel cylinders, a master brake cylinder, and inlet/outlet valves. As illustrated in Fig. 6 which represents the hydraulic brake structure, a spring and a piston are used to model the wheel cylinder

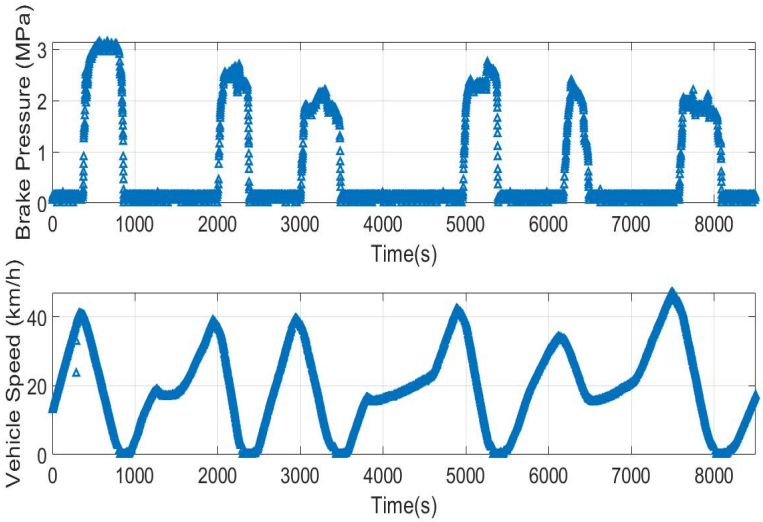


Figure 4.5: The vehicle speed and corresponding brake pressure

movements. Based on the hydraulic valve dynamics and fluid flow, the pressure of the wheel cylinder can be expressed by equation 20. Detailed models can be found in :

$$\dot{p}_{FW} = \frac{k_{FW}}{\pi^2 r_{FW}^4} C_d A_v \sqrt{\frac{2 \cdot \Delta p}{\rho_{fluid}}} \tag{4.17}$$

where r_{FW} and k_{FW} denote the radius of the piston and stiffness of the spring, respectively. A_v is the cross section area of the valve opening and C_d is the flow coefficient. Δp and ρ_{fluid} are the pressure difference across the valve and the density of the hydraulic fluid.

4.3.2 Data collection and preprocessing

The vehicle was run through several random driving cycles, giving a total of 10000 seconds of data. Vehicle states and powertrain data were collected using the CAN bus, at a frequency of 100 Hz. In order to enhance the training performance, the raw unbalanced features data were smoothed and subsequently scaled from 0 to 1 to reduce the effect of the dissimilar units of the used signals. Fig. 4.5 shows an example of the collected raw data of the vehicle speed and corresponding brake pressure.

4.3.3 Process of feature selection

Selecting unique, and redundancy-free features contribute to a successful training of the state estimator model. The main states of the vehicle and powertrain were selected to train the model of brake pressure state estimator, while the measured brake pressure value is used as a ground truth. In addition to the vehicle and powertrain parameters, the motor speed and torque, the battery voltage and current, and the state of charge (SoC) of the battery are used as unique features. This is because when the EV is in the deceleration mode, the electric motor works as a generator, recapturing the kinetic energy. This causes the motor and battery current to change from positive to negative, which indicates that the battery is being recharged by regenerated energy from braking. The mean and standard deviations of some vehicle states are also added as features.

4.4 Deep-Learning Based State Estimator Model

In this section, the proposed deep-structured neural network is evaluated for the purpose of brake pressure state estimation. The scaled EV features along with the ground truth brake pressure are considered for the training process of the DNN. The proposed DNN uses the vehicle states and powertrain states and the ground truth brake pressure values as target outputs while training the DL state estimator model. Fig. 4.5 illustrates some key features from the training data for one natural driving cycle.

Designing an accurate DL-Based state estimator model to estimate the brake pressure is not straightforward. Obtaining an accurate model is linked to the accuracy of the utilized training datasets, the structure of the network, training and optimization techniques. In this study, innovative deep learning techniques are exploited to enhance the training process of the DNN. Dropout regularization and Rectified Linear Units (ReLUs) are implemented to improve the prediction of the Brake pressure state estimator's model. As shown in Fig. 4.7, the proposed DNN consists of visible input and output layers and 3 hidden layers, along with the visible inputs and outputs layers.

The ReLU activation function is used to generate the output signal of the hidden values, it can be expressed as follows:

$$f(n) = \max(0, n) \tag{4.18}$$

The ReLU activation function is used to ensure accurate training for all hidden layers' nodes. Dropout regularization method is implemented to reduce the training cycle's

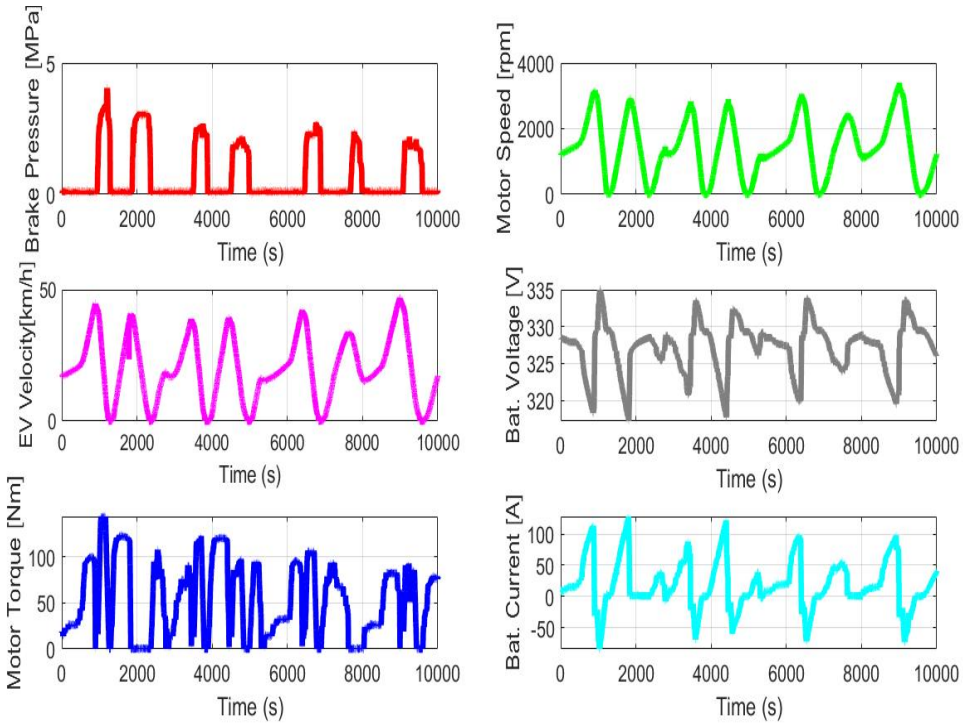


Figure 4.6: Sample of pre-scaled key features training data.

computations by considering arbitrarily selected nodes for the training rather than the entire net. This means that the dropped-out units have temporally no contribution on the forward path and any weight updates are not applied to the neuron on the backward path. The implementation of dropout is not cumbersome; it is based on picking random units with a predefined probability P to be excluded from the training process. A low probability value has small impact and high values may cause under-learning by the network. However, with respect to the size of the DNN and number of units, a probability value of 10%-50% can provide good performance.

4.5 Experimental Results and Discussions

This section presents the training and testing results of the implemented DL-based approach for the brake pressure state estimation. Discussions of dropout probability tuning are also included.

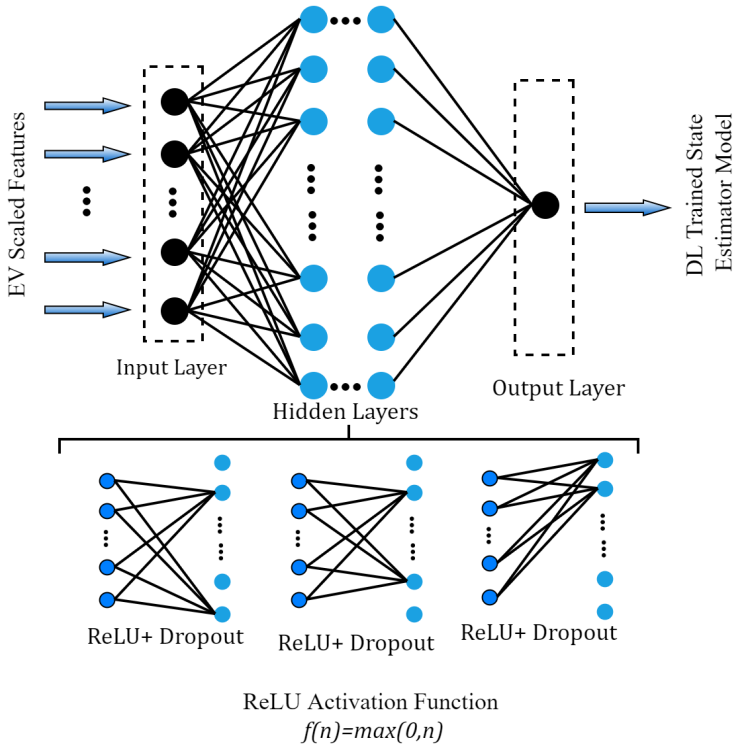


Figure 4.7: DNN Training Scheme

4.5.1 State estimator model training results

The proposed brake pressure state estimation method is implemented in Python with Keras. Several DL models were developed and trained by importing the experimental data to the Keras environment. A DL structure of 3 hidden layers with 60, 40 and 20 neurons respectively, was chosen based on its smallest MSE of 0.087. All the experiments shown were run on an AMD Radeon™ HD 6800 Series GPU. The ADAM optimization algorithm is used to update the weights and biases based on the Mean Squared Errors (MSE) loss function:

$$MSE = \frac{1}{2\mathrm{N}} \sum_{k=1}^{n} \left(Y_k - \widehat{Y}_k \right)^2 \tag{4.19}$$

where Y_k and $\widehat{Y}_k$ are the target and evaluated network outputs; respectively, N represents the number of training data points.

Dropout technique can be implemented for the hidden layers as well as for the visible layers. In order to investigate the best implementation with appropriate probability p, several batch training tests are performed with probabilities ranging from 0.1 to 0.5. The tests are performed with dropout being applied to both the visible input layer alone, and to the visible input and the hidden layers. Fig. 4.8 exhibits the average RMSE values over 200 epochs for both cases. As can be seen, the RMSE values increases as the dropout probability increases. It can be noticed that dropout is more feasible to be incorporated for hidden and visible input layers, based on the lower RMSE values. The dropout probability can be optimized through cross validation.

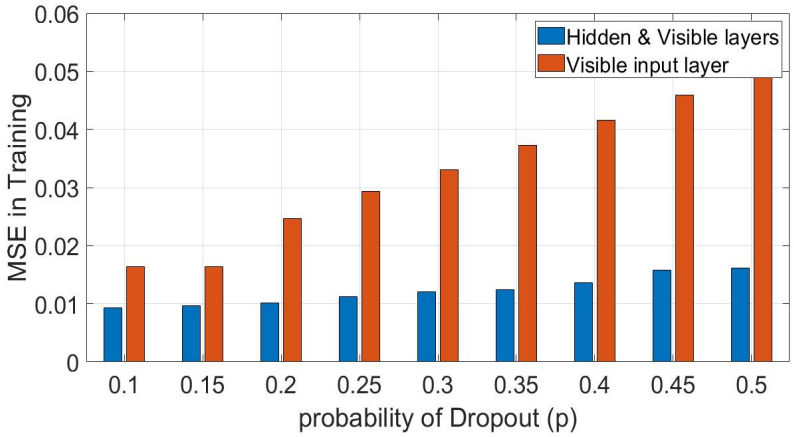


Figure 4.8: RMSE in validation with different dropout probabilities

The proposed model is trained using 8500 experimental data points over 200 epochs. Fig. 4.9 illustrates the accuracy of the training performance of the DNN. As shown, the losses represented by MSE values decrease as the DNN's weights and biases updated. This proves the accurate update of the weight and biases while training the DNN. Based on the small MSE values, it can be concluded that the training process has obtained an expressive neural network model based on the training set used.

4.5.2 State estimator model testing

In this section, the proposed DL-Based state estimation algorithm is tested and evaluated. A testing environment is developed for the trained model. A 20 % of the collected data points was selected for testing the proposed state estimator model. The testing data includes the vehicle states and the power train states. The output of the proposed DL-based

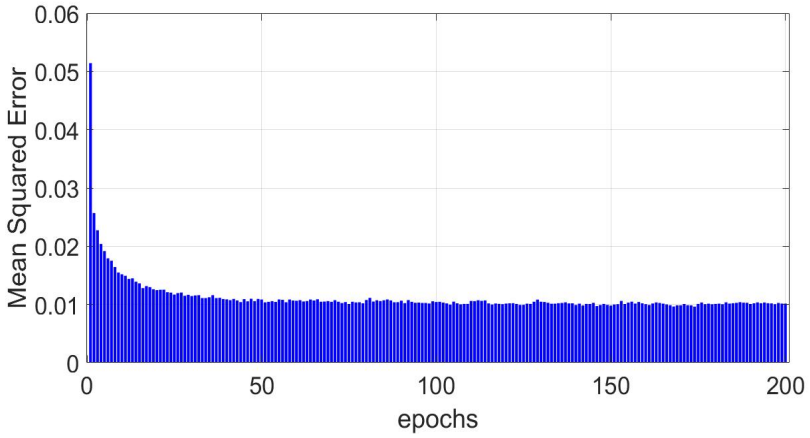


Figure 4.9: MSE in training over 200 epochs

state estimator has been compared to the ground truth values of the brake pressure. Fig. 4.10 shows the scaled brake pressure state estimation. The x-axis represents the number of samples while the y-axis represents the scaled brake pressure. As can be observed, the results of the proposed DL-based state estimation approach show a significant coincidence with the ground truth values of the brake pressure.

Figure 4.11 shows the state estimation error magnitudes. To evaluate the accuracy of the proposed approach, regression performance errors are quantified using two main indices, namely, the RMSE and the coefficient of determination R^2. R^2 is a measure of the model's predication accuracy. It falls between 0 and 1, and the higher the value of the coefficient R^2, the better the model at predicting the observations. R^2 is described as

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2} \tag{4.20}$$

where $\hat{y}$ represents the predicated values of the state y and $\bar{y}$ represent the mean value of the state.

The proposed model has achieved a regression accuracy with R^2 of 0.994, indicating that the proposed model of the brake pressure state estimator can achieve high predication-accuracy of the brake pressure. Compared with the conventional training technique [51], the proposed has achieved more accurate state estimation with RMSE of 0.048 MPa. Based on the presented results, the proposed method demonstrates an accurate sensor-independent brake pressure state estimation.

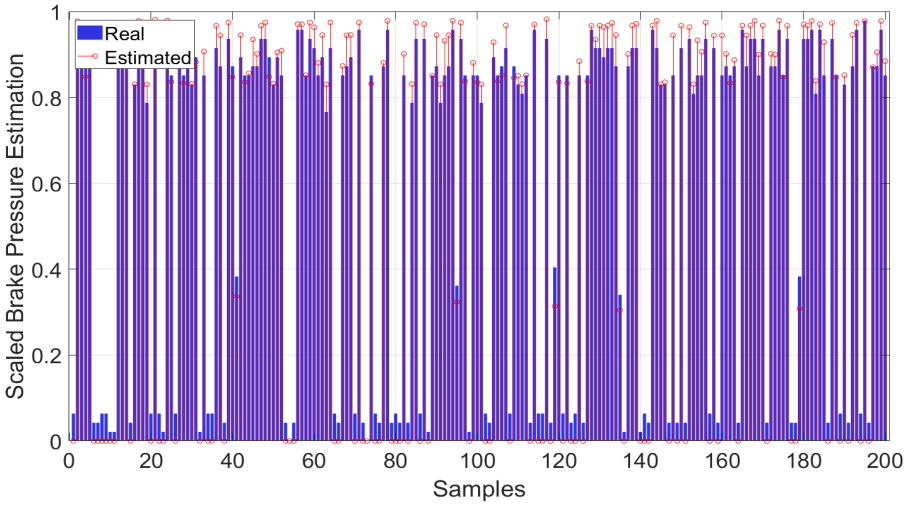


Figure 4.10: DL-Based Brake Pressure state estimation

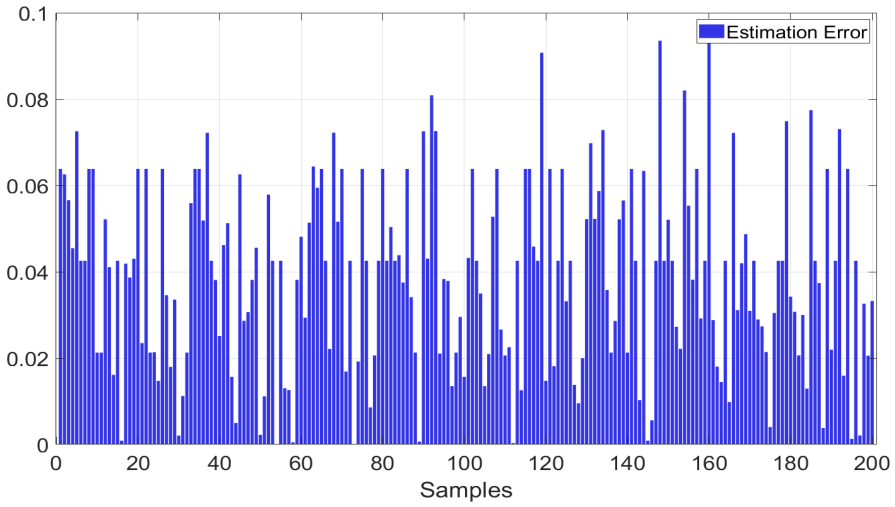


Figure 4.11: Error values of the proposed state estimation method

4.6 Conclusion

This Chapter proposes a novel DL-based state estimation algorithm for a safety-critical cyber-physical system. Using dropout and other DL modern elements such as ReLU activation functions, a novel DL-based model is introduced for brake pressure state estimation purposes. Real experiments for data collection are conducted via testing the EV on a chassis dynamometer under random driving cycles. The obtained data of the powertrain systems and vehicle states, as well as the ground truth values of the brake pressure are used for the training process. The DL-based state estimator model is trained using dropout with different probabilities. The training results show a high fitting accuracy as well as the testing results demonstrate the applicability and effectiveness of the proposed brake pressure state estimation approach.

As a future work of this research, the proposed state estimation can be further implemented and integrated with the onboard brake control system. The proposed DL-based model can also be flexibly expanded to estimate other states of the vehicle under several road conditions in urban and high-way scenarios. Furthermore, the estimated brake pressure information can be utilized in designing decision-making schemes for optimal braking in complex urban multiagent driving environment.

Chapter 5

Hierarchical Reinforced-learning for Feasible Decision-Making

Reinforcement learning-based techniques, powered by deep-structured neural nets, have demonstrated superiority over rule-based methods in terms of making high-level behavioral decisions due to qualities related to handling large state spaces. Nonetheless, their training time, sample efficiency and the feasibility of the learnt behaviors remain key concerns. In this Chapter, we propose a novel hierarchical reinforcement learning-based decision-making architecture for learning left-turn policies at unsignalized intersections with feasibility guarantees. The proposed technique is comprised of two layers; a high-level learning-based behavioral planning layer which adopts soft actor-critic principles to learn high-level, non-conservative yet safe, driving behaviors, and a low-level Model Predictive Control (MPC) framework to ensure feasibility of the two-dimensional left-turn maneuver. The high-level layer generates reference signals of velocity and yaw angles for the ego vehicle taking into account safety and collision avoidance with the intersection vehicles, whereas the low-level planning layer solves an optimization problem to track these reference commands taking into account several vehicle dynamic constraints and ride comfort. We validate the proposed decision-making scheme in simulated environments and compare with other model free Reinforcement Learning (RL) baselines. The validation results demonstrate that the proposed integrated framework possesses better training and navigation capabilities compared to the RL decision-making without integration.

The rest of the Chapter is structured as follows. Section 5.1 presents the proposed integrated decision-making scheme. Section 5.2 illustrates the experimental setup design and implementation details. Section 5.3 presents the results of the proposed integration followed by a discussion. Finally, section 5.4 concludes the proposed and future works.

5.1 A Hierarchical Reinforced-Learning Approach

The proposed approach has two primary planning layers: a high-level behavioral layer and a low-level motion control layer. The high-level behavioral planning is considered as a reinforcement learning problem, whereas the low-level motion is planned using a nonlinear model predictive control technique. The behavioural layer maps the observations collected from the driving environment into reference control signals, which are subsequently passed to MPC, which solves the tracking optimization problem and provides low-level commands to the ego vehicle. In this section, we show our multi-layered decision-making method for left-turn maneuvers at four-way unsignalized intersections. We focus on problem formulation, reward function design, and observation and action spaces. We next describe the implemented soft actor-critic principles, as well as the low-level motion planning and control, which is also presented in greater detail.

5.1.1 Overview of the Integrated Framework

We tackle a targeted driving scenario at an unsignalized intersection for the proposed decision-making challenge. The decision-making of the ego vehicle is coupled to the motion trajectory of the target vehicles while traversing the unsignalized intersection. This scenario simulates an ego vehicle navigating in a complex world where the simulated target vehicles do not decelerate or yield to the ego vehicle. We further assume that the target vehicle motion is observed by the ego vehicle [30].

Fig. 5.1 depicts the integrated decision-making framework. The decision-maker (agent), the ego vehicle, receives the perceptual observations and chooses actions accordingly. These actions are applied to the driving environment, and the environment returns rewards and new set of observations. As a standard RL setting, through these interactions with the driving environment, the ego-vehicle trains a policy that provides actions to maximize the future rewards. In our example, this policy is trained with the SAC algorithm to output reference velocity v_{ref} and heading signals θ_{ref}. The motion planning layer takes these reference signals as inputs to the two-dimensional tracking control problem, solving the formulated optimization problem while accounting for real-world constraints related to vehicle dynamics, urban traffic rules, and ride comfort. The optimized, feasible, control inputs are then produced to drive the vehicle's physical model in the simulated driving environment. The design process of these planning layers are explained in greater details in sections 5.1.2 and 5.1.3.

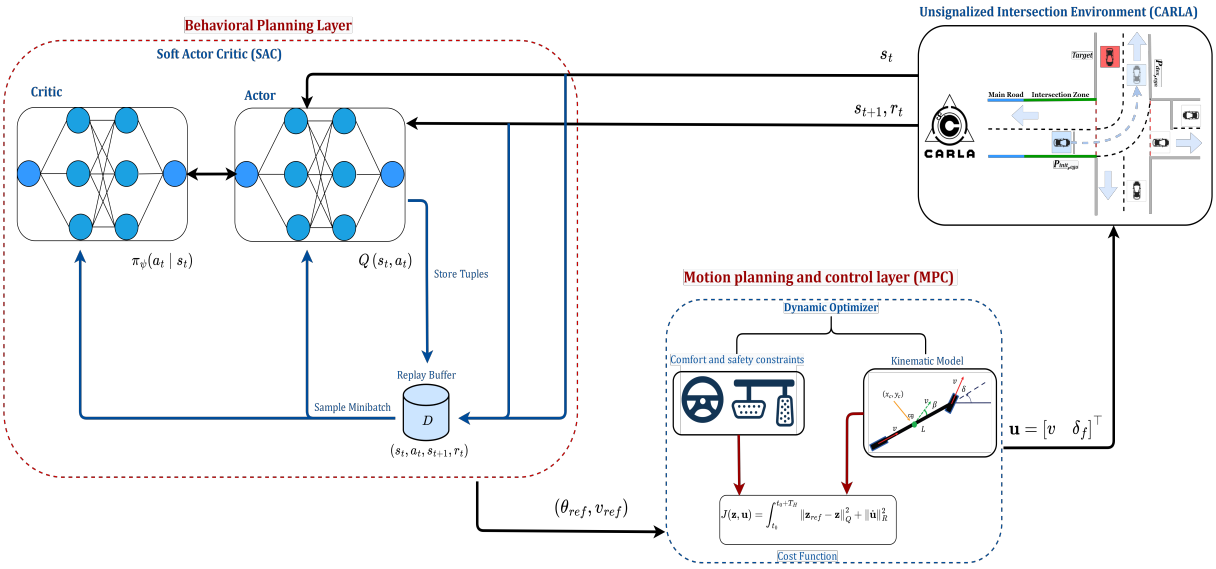


Figure 5.1: An illustrative sketch of the proposed hierarchical decision-making algorithm. The agent (decision maker) is denoted by the two integrated planning layers, whereas the CARLA simulated driving scenario is the environment.

5.1.2 High-level Behavioral Layer

A Markov Decision Process is used to formulate the left-turn behavioral planning problem. An MDP is described as a tuple $\{S, A, R, T, \gamma\}$. The observed state S includes the ego vehicle state as well as the state of the target vehicle. Specifically, ego and target velocities, positions, and information related to lane geometry are included. The transition function T maps state-action pairs to a new state. The immediate reward is defined by the reward function R, whereas γ represents the discount factor for long-term rewards. While abiding by the speed limit, we design our learning scheme based on minimizing the left-turn time and avoiding possible collision with other intersection vehicles. Hence, the optimality of the approach can be determined by identifying the best trade-off between these two conflicting interests.

Reward Function

The reward function requires a significant amount of shaping to show effective learning capabilities. Given that safety is the most important factor in autonomous driving, we formed

a reward function that prioritizes safety while maintaining a balance between transportation efficiency and safety during the left-turn maneuver. For the high-level two-dimensional left turn behavioral planning, the desired driving behavior of the learning agent is to proceed to the end of the route (completing the left turn) as efficiently as possible while remaining safe (in lane and no collisions). We performed extensive experimentation with several reward designs and found that a progress-along-route reward did not lead to efficient route completion because the agent was encouraged to take many steps to gather more reward. Instead, a negative reward for distance-to-goal did encourage efficient route completion. Furthermore, a positive reward is given for high speeds that remain under the speed limit of 12m/s, while negative rewards are given for exceeding the speed limit. A negative reward is given with magnitude proportional to lateral deviation from the lane center. The developed reward function is described as follows:

$$r_{ego} = r_{eff} + c_1 r_{dtg} + c_2 * r_{lat} + r_{terminal} \tag{5.1}$$

where r_{dtg} and r_{lat} represent the distance-to-goal and lateral deviation penalties, respectively. These terms are tuned by the constants c_1 and c_2. $r_{terminal}$ is the route completion/non-completion reward/penalty, which was tuned to reward the agent if the route was successfully completed or penalise the agent if there was a collision or exceeded the maximum number of steps. The remaining term r_{eff} is designed to ensure efficient left-turn crossing where the agent is encouraged to drive as quickly as possible while remaining under the speed limit v_{lim}. This term can be described as follows:

$$r_{eff} = \begin{cases} c_3 * (v_{ego} - v_{lim}) & if \quad v_{ego} > v_{lim} \\ c_4 * v_{ego} & otherwise \end{cases} \tag{5.2}$$

where v_{lim} and v_{ego} represent the urban speed limit and the current ego vehicle speed, respectively. $0 \geq c_3$ is a hyperparameter for adjusting the progression penalty when $(v_{ego} \geq v_{lim})$, whereas $1 \geq c_4 \geq 0$ is the positive progression reward.

Observation and Action Spaces

As we assume that the target vehicle's crossing behavior is observed by the ego vehicle, the learning agent has some information about other involved agents moving within its visibility range. Hence, the observation space includes information about the dynamic states of the ego vehicle itself and the target vehicle velocity, assuming that it is traversing

Table 5.1: Description of the observation and action states.

	Observation space ($\mathbb{R}^{15}$)	
v_e	magnitude of linear velocity of the ego vehicle	$\mathbb{R}$
$\dot{v}_e$	magnitude of linear acceleration of the ego vehicle	$\mathbb{R}$
$d\theta_e$	delta yaw angle of ego vehicle	$\mathbb{R}$
$\dot{\theta}_e$	yaw rate ego vehicle	$\mathbb{R}$
d_{cl}	lateral deviation from lane center of the ego vehicle	$\mathbb{R}$
a_{t-1}	previous action	$\mathbb{R}^2$
v_{tar}	linear velocity of the target vehicle	$\mathbb{R}$
p_{tar}	longitudinal and lateral distance to the target vehicle	$\mathbb{R}^2$
ψ_l	yaw angle of the lane at 1, 5, 10 meters ahead	$\mathbb{R}^3$
	Action space ($\mathbb{R}^2$)	
v_{ref}	reference velocity signal	$\mathbb{R}$
θ_{ref}	reference heading signal	$\mathbb{R}$

the intersection maximum urban speed allowed, and other input features related to the intersection geometry and collision flags. The state provided to the agent contains the ego velocity, acceleration, yaw angle delta with respect to the lane center, yaw rate, and the lateral deviation from lane center of the ego vehicle. The state also contains the yaw angle of the lane at intervals of 1, 5, and 10 meters in front of the ego, which is necessary to track the lane center. The state also contains the relative lateral and longitudinal distance between the target and ego vehicles, as well as the velocity of the target vehicle. Finally, the previous action taken is also recorded in the next state. All dynamic features in the observation vector have been normalized to ensure that they vary in identical ranges for improved convergence capabilities. Table 5.1 shows the definitions of the observation and action spaces.

Left-turn behavioral planning using SAC

To train the policy network, we implement SAC algorithm with adaptive exploration capability. Combining the actor-critic principle and adaptive entropy regularization, SAC trains its stochastic policy in an off-policy fashion to identify an optimal trade-off for the explore-exploit problem avoiding possible premature convergence. In addition to learning a

policy π_ψ, the algorithm learns two Q-function approximators (networks) Q_{ϕ_1}, Q_{ϕ_2}, which are updated by the following loss function:

$$\mathcal{L}\left(\phi_i\right) = \underset{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}}{\mathbb{E}} \left[\left(Q_{\phi i}\left(s_t, a_t\right) - y\left(r_t, s_{t+1}\right)\right)^2\right], \tag{5.3}$$

where i =1, 2 and $\mathcal{D}$ represents the experiences obtained by the agent's while exploring the environment. The target $y\left(r_t, s_{t+1}\right)$) is given by:

$$y\left(r_t, s_{t+1}\right) = r(s_t, a_t) + \gamma T_{tar}(s_{t+1}), \tag{5.4}$$

where T_{tar} function is optimized by the following equation:

$$T_{tar}(s_{t+1}) = \left(\min_{j=1,2} Q_{\phi_{\mathrm{tar},j}}\left(s_{t+1}, \tilde{a}_{t+1}\right) - \alpha \log \pi_\psi\left(\tilde{a}_{t+1} \mid s_{t+1}\right)\right), \tag{5.5}$$

where $\tilde{a}_{t+1}$ represent the next actions which are sampled from the policy $\pi_\psi\left(\cdot \mid s_{t+1}\right)$. α is used the tune exploitation-exploration trade-off which is governed by the policy's entropy. For instance, decreasing α values results in less exploration [56].

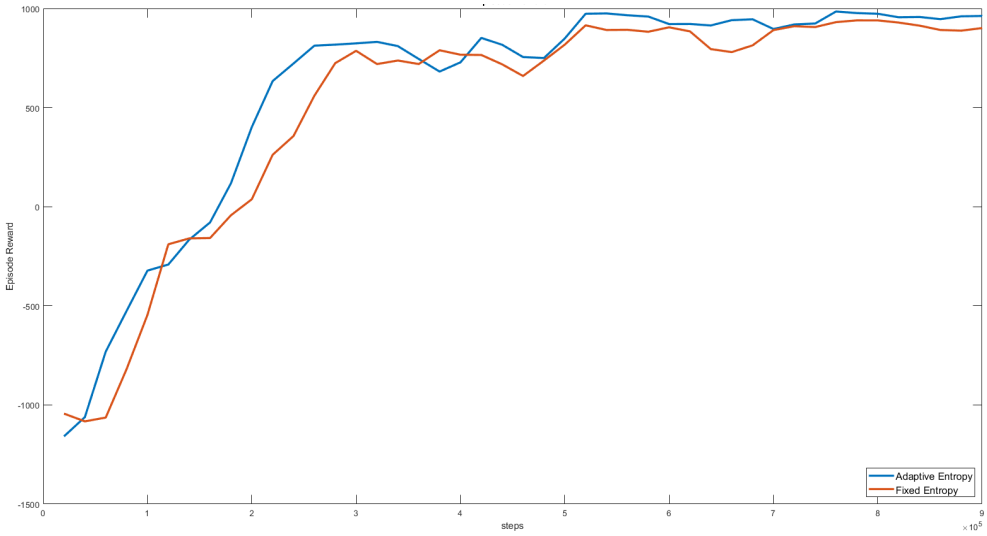
One implementation detail that we found effective was to schedule the entropy bonus α used in SAC based on the success rate of the task. Specifically, we use the following equation to adjust the exploration-exploitation term based on the performance of the agent:

$$\alpha = clip(1 - \lambda, v_1, v_2), \tag{5.6}$$

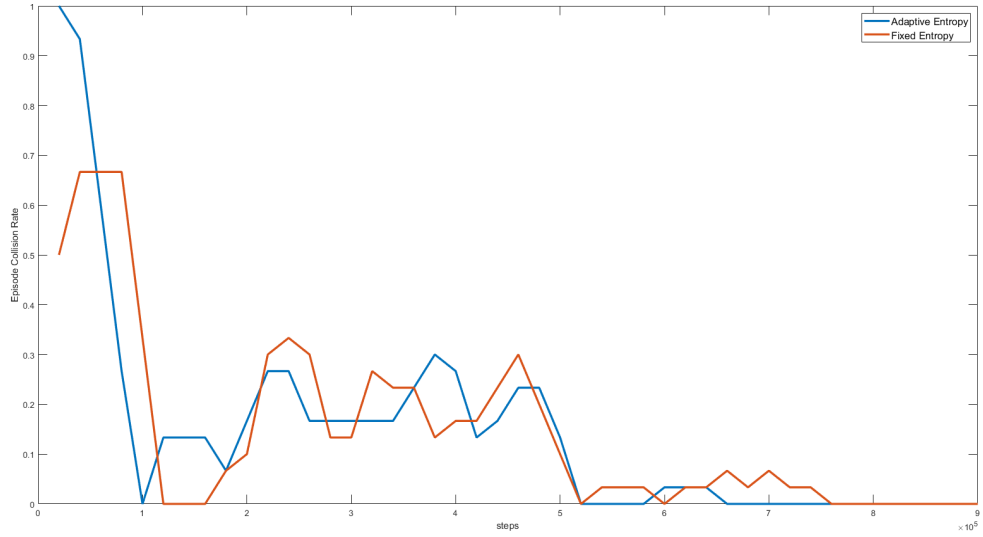
where λ is the success rate over the past 10 evaluation episodes, and the hyperparameters v_1 and v_2 are tuned to be 0.1 and 0.3, respectively. We start with $\alpha = 0.3$ and to ensure that the agent sufficiently explores the environment avoiding any local optimum. As shown in Fig. 5.2, adaptive entropy regularization has resulted in faster convergence, higher mean rewards and less collision rates.

The policy is learnt by maximizing the expected future return and expected future entropy as denoted in $V^\pi(s_t)$ function:

$$\begin{aligned} V^\pi(s_t) &= \underset{a_t \sim \pi}{\mathrm{E}} \left[Q^\pi(s_t, a_t)\right] + \alpha H(\pi(\cdot \mid s_t)) \\ &= \underset{a_t \sim \pi}{\mathrm{E}} \left[Q^\pi(s_t, a_t) - \alpha \log \pi(a_t \mid s_t)\right]. \end{aligned} \tag{5.7}$$



(a) Average rewards



(b) Collision rates

Figure 5.2: Comparison of training curves of driving polices (with and without adaptive entropy regularization).

Following the current policy, the actions can be derived from the current policy $\tilde{a}' \sim \pi_\psi(\cdot \mid s)$, whereas the states are drawn from the replay buffer $s \sim \mathcal{D}$. We use the reparameterization trick to optimise the policy $\pi_\psi(\cdot \mid s)$, in which a sample from actions is drawn by computing a deterministic function of state, policy parameters, and independent noise ξ as follows:

$$\tilde{a}_\psi(s) = \tanh\left(\mu_\psi(s) + \sigma_\psi(s) \odot \xi\right), \xi \sim \mathcal{N}(0, I) \tag{5.8}$$

where tanh is used to bound the obtained actions to a finite range of $[-1, 1]$.

The loss function for SAC learning, can be formulated as to maximize the expected future rewards plus the expected future entropy as can be described below:

$$\begin{aligned}\mathcal{L}(\psi)_{SAC} =& \mathbb{E}_{s_t \sim \mathcal{D}}\left[\alpha \log \pi_\psi\left(\tilde{a}_\psi(s_t) \mid s_t\right)\right. \\ &\left. - \min_{i=1,2} Q_{\phi_i}\left(s_t, \tilde{a}_\psi(s_t)\right)\right]\end{aligned} \tag{5.9}$$

5.1.3 Low-level Motion Planning and Control layer

The low-level layer is based on Model Predictive Control (MPC), that is responsible for vehicle motion control. The MPC controller respects the vehicle non-holonomic constraints by utilizing vehicle kinematic model for prediction [67]. The model is shown in Fig. 5.3 and stated below.

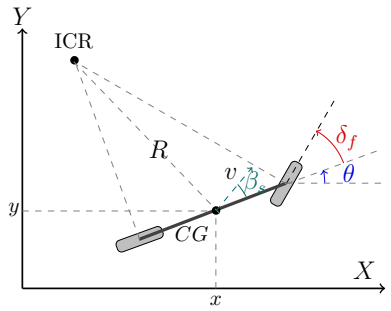


Figure 5.3: Kinematic bicycle model schematic.

$$\dot{\mathbf{x}}(t) = \begin{bmatrix}\dot{x} \\ \dot{y} \\ \dot{\theta}\end{bmatrix} = f\Big(\mathbf{x}(t), \mathbf{u}(t)\Big) = \begin{bmatrix} v\cos(\theta + \beta_s) \\ v\sin(\theta + \beta_s) \\ \frac{v\cos(\beta_s)\tan(\delta_f)}{L}\end{bmatrix}, \tag{5.10}$$

where

$$\beta_s = \arctan\left(\frac{l_r \tan\delta_f}{L}\right) \tag{5.11}$$

where the vehicle state is $\mathbf{x} = \begin{bmatrix} x & y & \theta \end{bmatrix}^\top$, x and y are the position of the vehicle in X-Y global frame, and θ is the vehicle orientation in the global frame. Furthermore, $\mathbf{u} = \begin{bmatrix} v & \delta_f \end{bmatrix}^\top$ is the vector of control actions, v is the velocity of the ego vehicle at its C.G., and δ_f is the steering angle. In Fig. 5.3, β_s is the side-slip angle of the vehicle, l_r is the distance between the rear axle and the C.G., and L is the wheelbase length of the vehicle.

The intuition behind using a vehicle kinematic model for optimization is that the dynamic effects are negligible due to low speed driving in urban environments. In addition, to ensure ride comfort and safety, several hard constraints are set on the optimization variables as given in table 5.2.

Table 5.2: Constraints set on the optimization variables.

Parameter	Lower Bound	Upper Bound
$\dot{\mathbf{u}}(t)$	$\dot{\mathbf{u}}(t)_{min} = \begin{bmatrix} -3\ \mathrm{m/s^2} \\ -\frac{\pi}{3}\ \mathrm{rad/s} \end{bmatrix}$	$\dot{\mathbf{u}}(t)_{max} = \begin{bmatrix} 5\ \mathrm{m/s^2} \\ \frac{\pi}{3}\ \mathrm{rad/s} \end{bmatrix}$
$\mathbf{u}(t)$	$\mathbf{u}(t)_{min} = \begin{bmatrix} -2.25\ \mathrm{m/s} \\ -\frac{\pi}{3}\ \mathrm{rad} \end{bmatrix}$	$\mathbf{u}(t)_{max} = \begin{bmatrix} 12\ \mathrm{m/s} \\ \frac{\pi}{3}\ \mathrm{rad} \end{bmatrix}$

To let the high-level control layer drive the ego vehicle while abiding tovehicle constraints and respecting the static map, e.g. road network, the MPC objective function is formulated to minimize the following running costs:

1. Velocity error between reference and actual speed of the vehicle.
2. Heading error between reference and actual heading of the vehicle.
3. Change in control actions.

Velocity and heading errors are based on the reference generated by the high-level control layer. Therefore, the high-level agent learns to generate the appropriate heading reference and velocity reference, and the low-level MPC achieves them while satisfying vehicle constraints. Therefore, the objective function J is given by:

$$J\big(\mathbf{z}, \mathbf{u}\big) = \int_{t_0}^{t_0+T_H} \big|\big|\mathbf{z}_{ref} - \mathbf{z}\big|\big|_Q^2 + \big|\big|\dot{\mathbf{u}}\big|\big|_R^2 \tag{5.12}$$

where $\mathbf{z} = \begin{bmatrix} \theta & v \end{bmatrix}^\top$ and, $Q \in \mathbb{R}^{2\times2}$, $R \in \mathbb{R}^{2\times2}$, t_0 is the initial time, and T_H is the prediction horizon. Accordingly, the Optimal Control Problem (OCP) can be formulated as follows:

$$\min_{\mathbf{u}(t)} \quad J(\mathbf{z}(t), \mathbf{u}(t)) \tag{5.13a}$$

$$\text{s.t.} \quad \dot{\mathbf{x}}(t) = f\big(\mathbf{x}(t), \mathbf{u}(t)\big), \qquad \forall t \in [t_0, t_0 + T_H] \tag{5.13b}$$

$$\dot{\mathbf{u}}_{min}(t) \leq \dot{\mathbf{u}}(t) \leq \dot{\mathbf{u}}_{max}(t), \forall t \in [t_0, t_0 + T_H] \tag{5.13c}$$

$$\mathbf{u}_{min}(t) \leq \mathbf{u}(t) \leq \mathbf{u}_{max}(t), \forall t \in [t_0, t_0 + T_H] \tag{5.13d}$$

In order to transform the OCP into a NLP, temporal discretization was applied to the system dynamics over a finite prediction horizon ($N = 15$). Specifically, the Runge–Kutta method (RK4) was used with a time step of $T = 0.25s$. As proposed by Bock and Plitt in [182], multiple shooting was used to enforce the dynamics of the system using constraints, reducing the nonlinearity of the objective function especially in the latter steps of the prediction horizon. To solve the NLP, the interior point optimizer (IPOPT) [183] from the CasADi [184] software package was used.

For more details, the implementation of the proposed integration between the behavioral and motion planning layers is described with greater details in algorithm 1.

5.2 Experiments

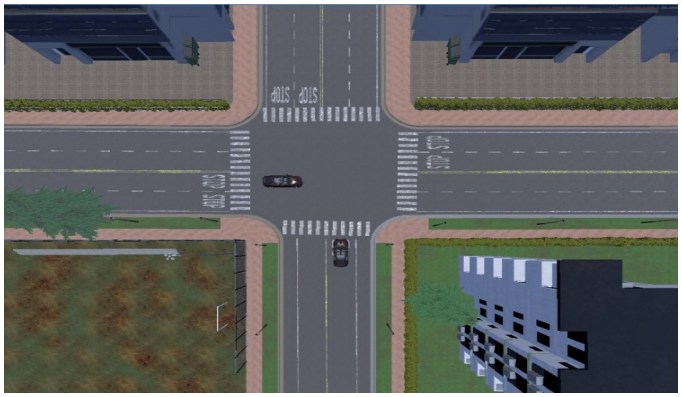
5.2.1 Environment Setup and Implementation details

We designed our reinforcement learning environment using the CARLA simulator [185] as it provides realistic simulated driving scenarios within urban environments, in addition to its flexible Python API. Fig. 5.4a, shows the task setup. We spawn the autonomous vehicle at a fixed predefined starting point aiming at learning how to follow the planned left-turn route efficiently and safely (no collision with target vehicle, no lane invasions), and reaching the predefined destination point. The target vehicle is spawned randomly covering all possible locations along the route as shown in Fig. 5.4b. We initiate up to 40 CARLA instances, simulating the same left-turn driving environment, which results in faster experience sampling compared to a single CARLA instance.

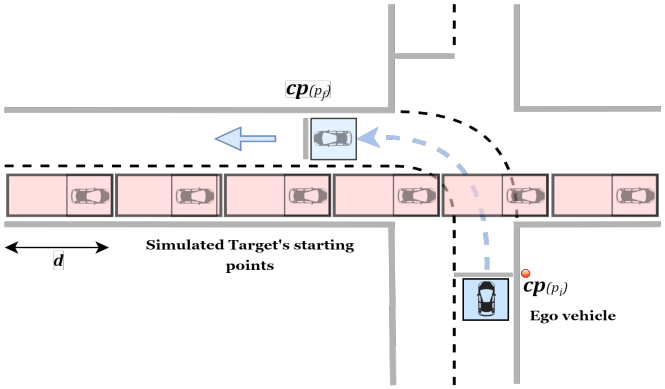
All the experiments in this study are conducted using NVIDIA GeForce RTX 3090. The GPU is well-equipped with 120 cores which enhance the performance of the neural networks used for policy optimization. The use of graphics processing unit is supported but not necessary for the training. We make use of Baidus deep learning framework-PaddlePaddle

Algorithm 1 SAC with MPC Integration

Initialize policy parameters θ, Q-function networks (Q_{ϕ_1}, Q_{ϕ_2}), empty replay buffer $\mathcal{D}$
for step in warm-up steps **do**
    Run network with randomized weights
end for
repeat
    Observe state s_t and sample action $\tilde{a}_\psi$
    Pass action $\tilde{a}_\psi$ to MPC as $\mathbf{z}_{ref}$
    MPC OCP is solved for optimal control actions while respecting vehicle dynamics and ride comfort using Eq.5.13
    Execute action $\mathbf{u}$ in the environment
    Observe next state s_{t+1} and obtain reward r_t using Eq.5.1 and Eq.5.2
    Increment total steps taken
    if Memory buffer $\mathcal{D}$ is full **then**
        Delete oldest transition in buffer
    end if
    Store transition (s_t, a_t, s_{t+1}, r_t) in memory buffer $\mathcal{D}$
    if time to update **then**
        Randomly sample mini-batch from replay buffer $\mathcal{D}$
        Update Q network policy parameters Q_{ϕ_1}, Q_{ϕ_2} using Eq.5.3 and Eq.5.4
        Update entropy bonus parameter α using Eq.5.6
    end if
    if s_{t+1} is terminal state **then**
        Reset environment
    end if
until Convergence or current step is equal to max steps



(a)



(b)

Figure 5.4: Driving experiment setup. Fig 5.4a illustrates the CARLA unsignalized intersection environment setup. Fig. 5.4b shows the the same setup with the possible starting points of the target vehicle.

which is available for both CPUs and GPUs. This forms the base of a high-efficient reinforcement learning framework, PARL [186]. This framework provides several research capabilities including reusability, reproducibility and extensibility. Building and integration of custom algorithms for policy training becomes feasible via use of this framework. It is composed of three major components - Agent, Algorithm and Model. The algorithm represents the update mechanism for parameters in model and therefore necessarily contains one model which abstracts the forward network. The model via abstraction delineates the forwards network defining critic or policy network which accepts states as input. The agent forms the data bridge for data I/O between the environment and algorithms. These three components serve as a compact API for distributed training by addition of a decorator.

Table 5.3: Experiment Parameters

Hyperparameter	Value
NN size	6X [256, ReLu]
Mini batch size	512
Replay buffer size	5e+05
Actor Learning Rate	3e-04
Critic Learning Rate	3e-04
Exponential Discount Factor	0.99
Max Episode Steps	500

Our customized decision-making reinforcement learning agent is implemented using PARL. We have chosen PARL due to its capability in supporting high-performance training parallelization with large number of CPUs and multi-GPUs, which is necessary to collect a large volume of experiences on the relatively slow Carla simulator. Additionally, PARL offers existing implementations of popular model free algorithms (TD3, PPO, SAC) on the Mujoco task set, which we were able to adapt to the CARLA-based environment we propose. The hyper-parameters used for the parallel version of SAC have been listed in Table 5.3 and the simulation timestep is 0.1 sec.

5.2.2 Policy training and evaluation

The SAC neural network is randomly initialized and trained via maximizing the left-turn reward function explained in (Eq. 5.1). We defined the stopping criteria for any training episode to be in cases where the agent exceeds the predefined maximum number of steps, reaches the destination, collides with the target vehicle, or lateral deviates from lane center more than 7.5 meters. The target vehicle is programmed to not yield to the ego vehicle as well as to abide by the urban speed limit.

To provide a holistic evaluation of the proposed design performance, we compare it to other model-free DRL techniques that can handle the continuous action space of the problem. As part of this experimental validation, we test the Twin Delayed Deep Deterministic Policy Gradient (TD3) and proximal policy optimization (PPO) approaches, which are common baselines for RL policy learning comparison in autonomous vehicles [56]. TD3

[187] is a modified, state of the art actor-critic, Deep Deterministic Policy Gradient algorithm for problems with continuous control domains. On the other hand, PPO is an improved on-policy of Trust Region Policy Optimization (TRPO) for robotics and games playing applications [188, 189].

5.3 Results and Discussion

In this section, we present the results of the proposed decision-making approach for left-turn traversing behaviour at four-way unsignalized Intersection. In section 5.3.1, we demonstrate the high-level behavioral planning layer performance using several model-free DRL algorithms, whereas the results of the integrated scheme are highlighted in section 5.3.2. We also conclude this section with remarks highlighted in section 5.3.3.

5.3.1 Model-free behavioral planning comparison

We compare the learning performance of SAC with other model-free RL algorithms, namely TD3 and PPO, in this section. For this comparison, the low-level motion planning is not integrated, the decision-maker (agent) receives observations from the intersection driving environment and maps them directly into throttle and steering commands executed by the environment. As seen in Fig. 5.5a, SAC outperforms both TD3 and PPO in terms of maximizing cumulative reward with fewer samples. SAC and TD3 do achieve a higher reward compared to PPO. This can be attributed to the entropy bonus in SAC allowing it to discover a better policy via exploration and double-critic architecture in SAC which improves the learning performance by reducing the overestimation bias. Furthermore, PPO requires the largest number of policy updates to converge but less consistently, as shown in Fig. 5.5b, where collisions occur with higher rate compared to SAC and TD3.

5.3.2 Integrated scheme Results

In this section, we illustrate the results of the proposed hierarchical decision-making scheme at unsignalized intersections. These results demonstrate the effectiveness of learning efficient, yet safe, left-turn behaviors with feasibility guarantees. The training performance of the network policy is evaluated using predefined Key Performance Indicators (KPIs). Among these, the average episodic reward, the average success rate, the average collision

rate and the max episodic speed. Fig. 5.6 shows the training evaluation for the first 500k training steps.

As shown in Fig. 5.6, at the very beginning of the training process, the average reward is noticeably low, which means that the agent is being heavily penalized as the collision rate is very high which hinders the agent from completing the task. As the training passes 30k steps, the agent significantly learns how to avoid colliding with the target vehicle as Fig. 5.6 shows a significant quick drop in the collision rate which align with the significant ascension of the average reward. However, the success rate still ranges between 30% to 40% agent which shows that the policy is not trained sufficiently. The policy starts its convergence after approximately 200k steps where the average reward converges to a high value and the average success rate fluctuates with values above 90%. Fig. 5.d illustrates the max episodic speed values during the training process. It can be discerned that the proposed decision-making approach provides feasible actions that abide by the constraints and ride comfort with a success rate above 90%.

The same training scenario, where the ego vehicle embarks on its two dimensional motion at the stop line and the target vehicle is spawned randomly in the scene, is adopted for testing the trained decision-making model. After conducting several training experiments for the developed integrated policy, a consistent performance is observed (see Fig. 5.8). We then save the superior trained policy to test the performance of the left-turn behavior over 1000 episodes [65]. The results show that the agent can maneuver left-turns with a success rate of 97.8%, colliding only once with the target vehicle, and failing to complete the task successfully in 21 coincidences (due to exceeding the maximum number of steps or lane departure). The effectiveness of the learnt traversal policy is visually demonstrated in Fig. 5.7, where we show a left-turn maneuver for one of the challenging traversal scenarios where the target vehicle shares conflict points with the ego vehicle. As seen in Fig. 5.7a, the ego is at the stop line waiting for the target vehicle to traverse the intersection environment. The non-cautious, yet safe, behavior of the trained policy can be shown in Fig. 5.7b, which is snipped after 1 sec, where the ego vehicle starts it is two-dimensional motion as quickly as safely possible. Fig. 5.7c and Fig. 5.7d represent the left-turn progress when nearly and fully completed, respectively.

5.3.3 Discussion on the Framework's Verification and Validation

In a recent review article on the verification and validation (V&V) techniques of decision-making and planning approaches in autonomous driving, the authors categorise the (V&V) approaches into three primary classes: fault injection testing, formal verification, and

scenario-based testing [190]. While fault injection approaches focus on examining the robustness of the decision-making schemes under software or hardware faults, formal verification methods examine the correctness of the developed decision making scheme from a logical and mathematical standpoint. Scenario-based testing approaches, On the other hand, are focused on safety, and are classified into fundamental and advanced approaches based on the interaction between road users in generated scenarios. As they address safety-critical scenarios and edge cases, standard RL and advanced RL-based decision-making techniques, such as DRL and adversarial approaches, fall under the category of advanced scenario-based testing methodologies.

Looking at the aforementioned results from a safety standpoint which is represented by the discussed KPIs, such as collision rate and success rate, it can be noticed that learning driving policies with low-level motion planning integrated, is necessary for learning safety-critical driving behaviors with feasibility guarantees. With the MPC enabled, the policy start converging after converges after 200k training steps with success rate 100% as depicted in Fig. 5.6a and Fig. 5.6b, whereas the standalone SAC converges after 500k steps. This could be attributed to the fact that the control inputs produced by the MPC, to the environment, have been already optimized taking real-life driving constraints into account.

Comparing our results with the state-of-the-art published works, including random curriculum learning-based results provided in [30], we can discern that our work results in faster convergence and higher success rate.

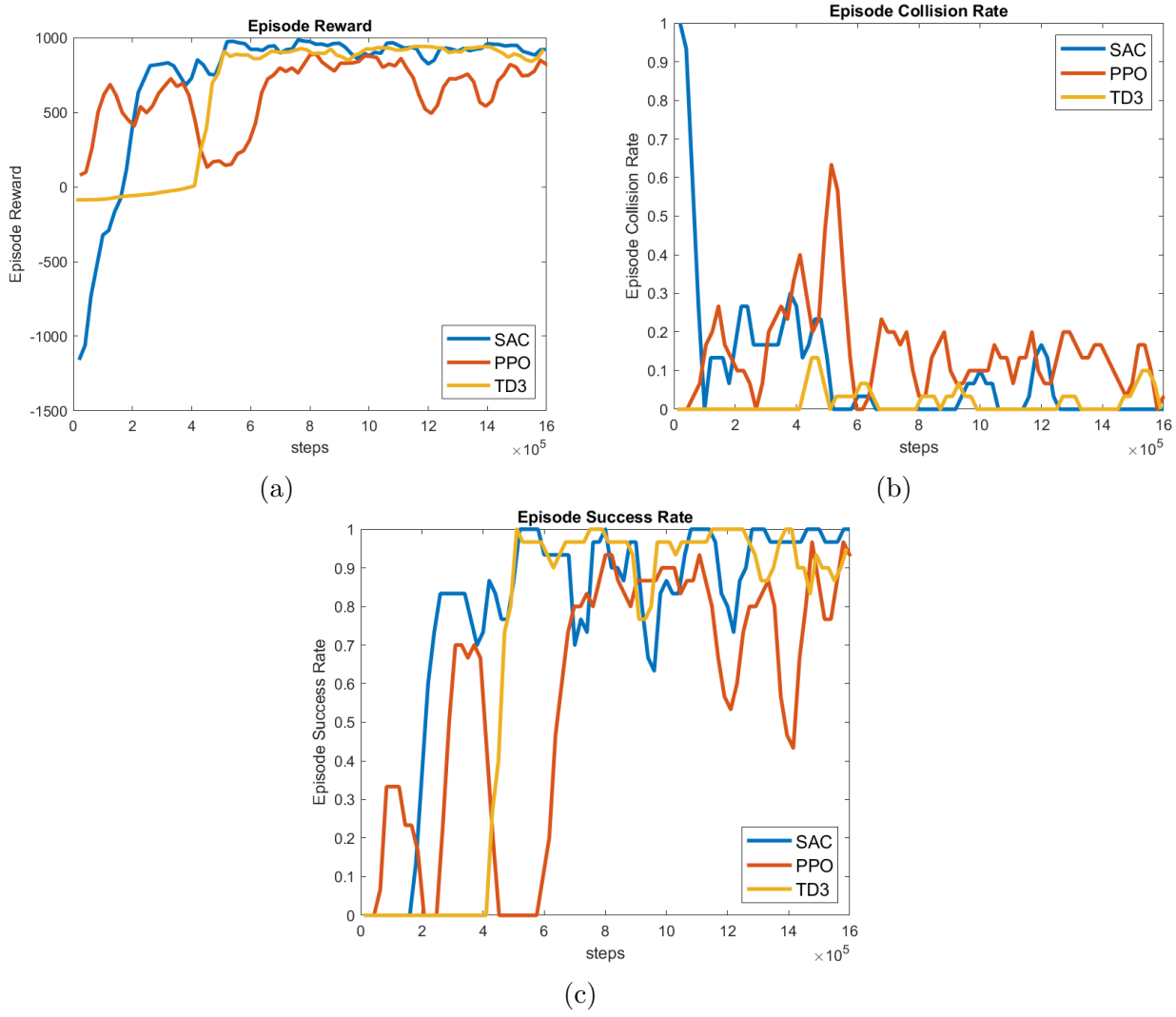


Figure 5.5: Model-free comparisons. Fig. 5.5a and and Fig. 5.5b show the average episodic reward and collision rate, respectively. Fig. 5.5c represent the average success rate (over 10 evaluation episodes). The models are trained until their performance converged.

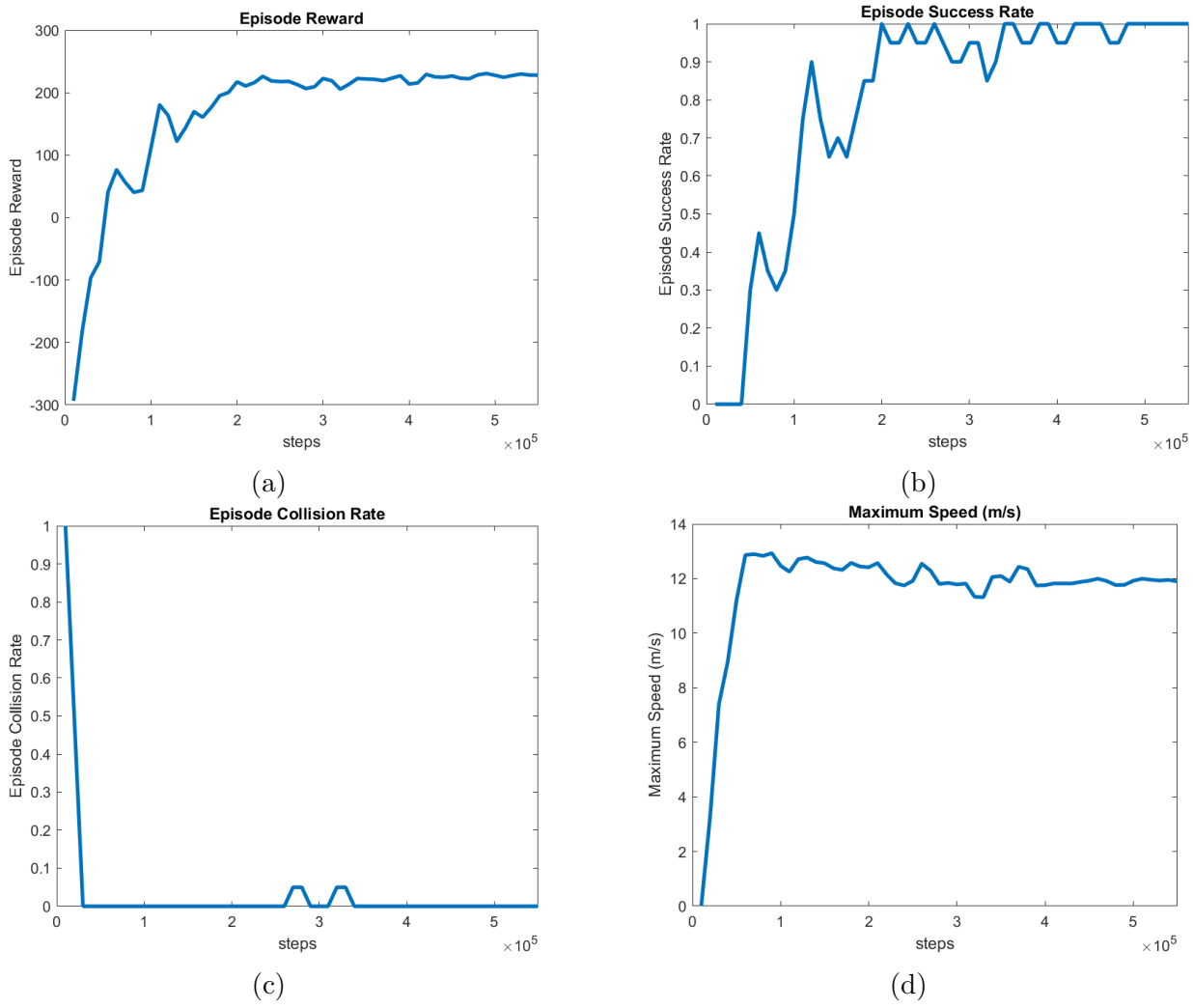
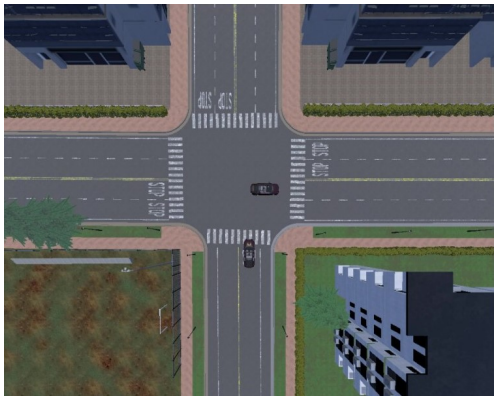
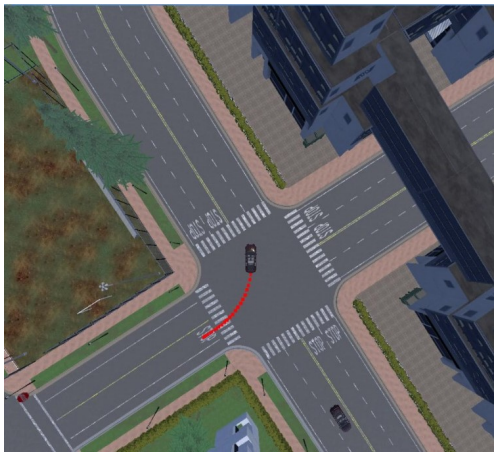


Figure 5.6: Training curves of the proposed reinforced learning hierarchical decision-making scheme. Fig. 5.6a illustrates the average episodic reward. Fig. 5.6b and 5.6c exhibit the success rate and the collision rate, respectively. The maximum episodic speed is plotted in Fig. 5.6d where the agent converges to the max urban speed allowed (12 m/s). The Policy is trained until convergence.

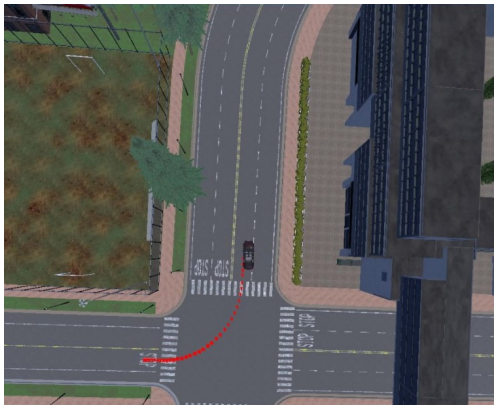
(a) T = 0 Sec



(b) T = 1 Sec



(c) T = 4 Sec



(d) T = 6 Sec

Figure 5.7: Testing snippets of left-turn maneuvers at unsignalized Intersection.

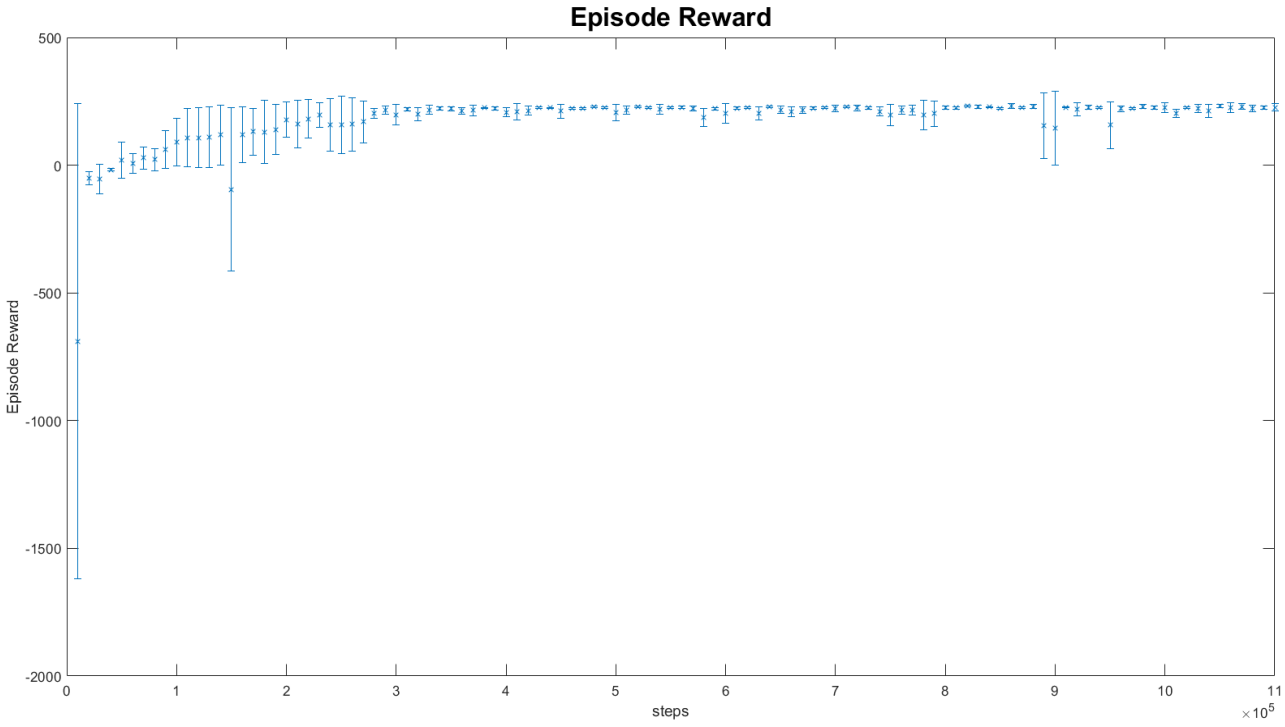


Figure 5.8: Mean of the average rewards (denoted by x) and standard deviations of three training experiments of the developed integrated model

5.4 Conclusion

In this Chapter, a novel hierarchical reinforcement learning-based decision-making scheme is proposed for automated unprotected left-turn maneuvers at unsignalized intersections. The proposed novel integrated scheme combines soft-actor-critic and model predictive control principles for high-level behavioral planning and low-level motion planning layers, respectively. The goal of this integration is to learn high-fidelity left-turn behaviours while accounting for real-world constraints related to vehicle dynamics, urban traffic rules, and ride comfort. For adaptive exploration-exploitation capabilities, we modify the SAC implementation by linking the entropy bonus updates to the agent's episodic success rate. A customized CARLA urban driving environment is designed to validate the proposed decision-making scheme. The high-level training comparison shows a superiority of SAC over other model-free learning schemes including TD3 and PPO. Moreover, the training results of the integrated framework illustrates the effectiveness of the proposed method in terms of performance and sample efficiency. Finally, the testing visual demonstration demonstrates the efficiency and safety of the learn left-turn behaviors yielding a success rate of 97.8% over 1000 testing episodes.

We should acknowledge, however, that there are few limitations of the proposed work. Among these, first, the generalization capability to handle environments with more agents coming from different directions and lanes, as well as the ability of handling noisy perceptual observations. Second, we need to improve the model's accuracy and navigation capabilities under occlusions where the intersection environment is partially observable. Therefore, we have future research directions towards improving the model in these aspects.

Chapter 6

Conclusions and Future Works

In this thesis, we develop high-fidelity learning-based frameworks for feasible automated decision-making in urban environments. Throughout this dissertation, we emphasize on practical and technical design considerations to train safety-critical, efficient with feasibility guarantees, decision-making policies in safety-critical urban environments. Based on our in-depth survey conducted in **Chapter 3**, we found that research efforts are still required to tackle the real-world challenges of unsignalized intersection-traversal problem. We suggest methods and heuristics that can be used to facilitate real-world driving for testing and validation purposes of the RL-based models. Taking one of the concluded recommendations into consideration, in **Chapter 4**, we design a learning-based sensor-independent state-estimation technique for cyber-physical system in urban environments. Furthermore, in **Chapter 5**, we propose a novel hierarchical reinforcement learning-based decision-making architecture for learning left-turn policies at unsignalized intersections with feasibility guarantees.

In further depth, in **Chapter 3**, we study the decision-making problem at urban critical environments, namely unsignalized intersections. We review the published works on various aspects related to decision-making challenges associated with decision-making at unsignalized intersections with a focus on learning-based schemes. We discuss these schemes in terms of the tackled driving scenario, the involved challenges, the proposed learning-based designs and the validation in simulations and real-world environments. We identify key remarks for better handling the large partially-observable state space of the problem. Moreover, based on our discussion and investigation, we found that research efforts are still required to tackle the real-world challenges of unsignalized intersection-traversal problem. More specifically, we found that the state-of-the-art decision-making approaches focus on advancing the high-level behavioral reasoning neglecting the importance of feasibility guar-

antees and provided by motion planning and low-level high-fidelity feedback controller and state estimator layer. Hence, we recommend two main research avenues that can be pursued to facilitate learning feasible behaviors and validating the learnt behaviors in real-life urban driving settings.

In **Chapter 4**, we developed a dropout-based training scheme for a safety-critical system state estimator in urban driving environment. For the proposed supervised learning scheme, real-life experiments were conducted to collect the ground truth values of the cyber-physical system and other measurements of the features vector including of the powertrain systems and vehicle states. The training and the testing results demonstrate the applicability and the superiority of the proposed estimation technique over the conventional training schemes. The proposed state estimation scheme is agnostic to decision-making in urban settings, but can be integrated with the onboard brake control system for high-fidelity motion planing purposes.

In **Chapter 5**, based on the recommendations concluded from the research study presented in **Chapter 3**, we develop a novel hierarchical reinforcement learning-based decision-making scheme for left-turn maneuvers at unsignalized intersections with feasibility guarantees. The proposed scheme incorporates soft-actor-critic and model predictive control principles for high-level behavioral planning and low-level motion planning layers, respectively. The goal of this integration is to learn high-fidelity left-turn behaviours while accounting for real-world constraints related to vehicle dynamics, urban traffic rules, and ride comfort. A customized CARLA urban driving environment is designed to validate the proposed decision-making scheme. The high-level training comparison shows a superiority of SAC over other model-free learning schemes including TD3 and PPO. Moreover, the training results of the integrated framework illustrates the effectiveness of the proposed method in terms of performance and sample efficiency. Finally, the testing results demonstrate the efficiency and safety of the learn left-turn behaviors yielding a success rate of 97.8% over 1000 testing episodes.

We then highlight the future research avenues of the proposed work. First, As mentioned in **Chapter 5**, the proposed decision-scheme is developed in simulated unsignalized environment. However, we highlighted in our survey presented in **Chapter 3** that validating the trained policies in real-world driving settings is an active area of research. We highlight that transfer learning approaches including Domain adaptation and Domain Randomization can be employed to facilitate testing in real-world driving settings. Second, given the limitations of the proposed decision-making approach highlighted in **Chapter 5**, we need to improve the model accuracy to ensure collision-free maneuvers. In addition, improving the model's generalization capabilities to handle environments with more participants traversing from different directions and lanes, as well as the ability of handling noisy

perceptual observations is a possible research direction. Third, with regard to the state estimation technique proposed in **Chapter 5**, the learning-based estimation scheme can be flexibly extended to estimate other safty-critical states under several road conditions in urban driving settings. Furthermore, the estimated states can be integrated with the developed decision-making scheme for high-fidelity motion planning and control in real-life driving settings.

References

[1] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, 2018.

[2] S. Broumi, A. Bakal, M. Talea, F. Smarandache, and L. Vladareanu, "Applying dijkstra algorithm for solving neutrosophic shortest path problem," in *2016 International conference on advanced mechatronic systems (ICAMechS)*. IEEE, 2016, pp. 412–416.

[3] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[4] Q. Liu, P. Hou, G. Wang, T. Peng, and S. Zhang, "Intelligent route planning on large road networks with efficiency and privacy," *Journal of Parallel and Distributed Computing*, vol. 133, pp. 93–106, 2019.

[5] J. Li, D. Fu, Q. Yuan, H. Zhang, K. Chen, S. Yang, and F. Yang, "A traffic prediction enabled double rewarded value iteration network for route planning," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4170–4181, 2019.

[6] A. Ahmad, S. Din, A. Paul, G. Jeon, M. Aloqaily, and M. Ahmad, "Real-time route planning and data dissemination for urban scenarios using the internet of things," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 50–55, 2019.

[7] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[8] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA urban challenge: autonomous vehicles in city traffic*. springer, 2009, vol. 56.

[9] M. Dikmen and C. M. Burns, "Autonomous driving in the real world: Experiences with tesla autopilot and summon," in *Proceedings of the 8th international conference on automotive user interfaces and interactive vehicular applications*, 2016, pp. 225–228.

[10] H. Wang, A. Khajepour, D. Cao, and T. Liu, "Ethical decision making in autonomous vehicles: Challenges and research progress," *IEEE Intelligent Transportation Systems Magazine*, 2020.

[11] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on control systems technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[12] S. Verma, Y. H. Eng, H. X. Kong, H. Andersen, M. Meghjani, W. K. Leong, X. Shen, C. Zhang, M. H. Ang, and D. Rus, "Vehicle detection, tracking and behavior analysis in urban driving environments using road context," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1413–1420.

[13] R. Senanayake, M. Toyungyernsub, M. Wang, M. J. Kochenderfer, and M. Schwager, "Directional primitives for uncertainty-aware motion estimation in urban environments," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–6.

[14] X. Xu, L. Zuo, X. Li, L. Qian, J. Ren, and Z. Sun, "A reinforcement learning approach to autonomous decision making of intelligent vehicles on highways," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[15] J. Liao, T. Liu, X. Tang, X. Mu, B. Huang, and D. Cao, "Decision-making strategy on highway for autonomous vehicles using deep reinforcement learning," *IEEE Access*, vol. 8, pp. 177 804–177 814, 2020.

[16] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications," *IEEE/ASME Transactions on mechatronics*, vol. 21, no. 2, pp. 740–753, 2015.

[17] S. Gilroy, E. Jones, and M. Glavin, "Overcoming occlusion in the automotive environment-a review," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[18] X. Huang, S. G. McGill, B. C. Williams, L. Fletcher, and G. Rosman, "Uncertainty-aware driver trajectory prediction at urban intersections," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9718–9724.

[19] E.-H. Choi, "Crash factors in intersection-related crashes: An on-scene perspective," Tech. Rep., 2010.

[20] M. Wayland. (2020) Gm's cruise begins testing autonomous vehicles without human drivers in san francisco. [Online]. Available: https://www.cnbc.com/2020/12/09/gms-cruise-begins-testing-autonomous-vehicles-without-human-drivers-in-san-francisco.html

[21] Y. Chen, J. Zha, and J. Wang, "An autonomous t-intersection driving strategy considering oncoming vehicles based on connected vehicle technology," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2779–2790, 2019.

[22] J. Hawke, R. Shen, C. Gurau, S. Sharma, D. Reda, N. Nikolov, P. Mazur, S. Micklethwaite, N. Griffiths, A. Shah *et al.*, "Urban driving with conditional imitation learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 251–257.

[23] "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles." SAE, 2016.

[24] Z. H. Khattak, M. D. Fontaine, and B. L. Smith, "Exploratory investigation of disengagements and crashes in autonomous vehicles under mixed traffic: An endogenous switching regime framework," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[25] S. Pruekprasert, J. Dubut, X. Zhang, C. Huang, and M. Kishida, "A game theoretic approach to decision making for multiple vehicles at roundabout," *arXiv preprint arXiv:1904.06224*, 2019.

[26] G. Li, S. Li, S. Li, Y. Qin, D. Cao, X. Qu, and B. Cheng, "Deep reinforcement learning enabled decision-making for autonomous driving at intersections," *Automotive Innovation*, vol. 3, no. 4, pp. 374–385, 2020.

[27] X. Huang, S. G. McGill, J. A. DeCastro, L. Fletcher, J. J. Leonard, B. C. Williams, and G. Rosman, "Diversitygan: Diversity-aware vehicle motion prediction via latent semantic sampling," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5089–5096, 2020.

[28] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 5–17, 2018.

[29] Y. Jeong and K. Yi, "Target vehicle motion prediction-based motion planning framework for autonomous driving in uncontrolled intersections," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[30] Z. Qiao, K. Muelling, J. M. Dolan, P. Palanisamy, and P. Mudalige, "Automatically generated curriculum based reinforcement learning for autonomous vehicles in urban environment," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1233–1238.

[31] M. Barbier, C. Laugier, O. Simonin, and J. Ibañez-Guzmán, "Probabilistic decision-making at road intersections: Formulation and quantitative evaluation," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 795–802.

[32] S. G. McGill, G. Rosman, T. Ort, A. Pierson, I. Gilitschenski, B. Araki, L. Fletcher, S. Karaman, D. Rus, and J. J. Leonard, "Probabilistic risk metrics for navigating occluded intersections," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4322–4329, 2019.

[33] Z. Qiao, K. Muelling, J. Dolan, P. Palanisamy, and P. Mudalige, "Pomdp and hierarchical options mdp with continuous actions for autonomous driving at intersections," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2377–2382.

[34] C. Li and K. Czarnecki, "Urban driving with multi-objective deep reinforcement learning," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 359–367.

[35] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Safe reinforcement learning with scene decomposition for navigating complex urban environments," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1469–1476.

[36] M. Bouton, J. Karlsson, A. Nakhaei, K. Fujimura, M. J. Kochenderfer, and J. Tumova, "Reinforcement learning with probabilistic guarantees for autonomous driving," *arXiv preprint arXiv:1904.07189*, 2019.

[37] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2034–2039.

[38] J. Chen, B. Yuan, and M. Tomizuka, "Deep imitation learning for autonomous driving in generic urban scenarios with enhanced safety," *arXiv preprint arXiv:1903.00640*, 2019.

[39] M. R. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, "Cooperative collision avoidance at intersections: Algorithms and experiments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1162–1175, 2013.

[40] T. Wu, M. Jiang, and L. Zhang, "Cooperative multiagent deep deterministic policy gradient (comaddpg) for intelligent connected transportation with unsignalized intersection," *Mathematical Problems in Engineering*, vol. 2020, 2020.

[41] Z. Wang, K. Han, and P. Tiwari, "Digital twin-assisted cooperative driving at nonsignalized intersections," *arXiv preprint arXiv:2105.01357*, 2021.

[42] R. Tian, N. Li, I. Kolmanovsky, Y. Yildiz, and A. R. Girard, "Game-theoretic modeling of traffic in unsignalized intersection network for autonomous vehicle control verification and validation," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[43] N. Li, Y. Yao, I. Kolmanovsky, E. Atkins, and A. R. Girard, "Game-theoretic modeling of multi-vehicle interactions at uncontrolled intersections," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[44] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller, "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1671–1678.

[45] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[46] A. Fazeli, M. Zeinali, and A. Khajepour, "Application of adaptive sliding mode control for regenerative braking torque control," *IEEE/ASME Transactions On Mechatronics*, vol. 17, no. 4, pp. 745–755, 2011.

[47] A. Dadashnialehi, A. Bab-Hadiashar, Z. Cao, and A. Kapoor, "Intelligent sensorless antilock braking system for brushless in-wheel electric vehicles," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1629–1638, 2014.

[48] J. J. Castillo, J. A. Cabrera, A. J. Guerra, and A. Simón, "A novel electrohydraulic brake system with tire–road friction estimation and continuous brake pressure control," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1863–1875, 2015.

[49] Y. Li, C. Tang, S. Peeta, and Y. Wang, "Integral-sliding-mode braking control for a connected vehicle platoon: Theory and application," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4618–4628, 2018.

[50] A. Dadashnialehi, A. Bab-Hadiashar, Z. Cao, and R. Hoseinnezhad, "Reliable emf-sensor-fusion-based antilock braking system for bldc motor in-wheel electric vehicles," *IEEE sensors letters*, vol. 1, no. 3, pp. 1–4, 2017.

[51] C. Lv, Y. Xing, J. Zhang, X. Na, Y. Li, T. Liu, D. Cao, and F.-Y. Wang, "Levenberg–marquardt backpropagation training of multilayer neural networks for state estimation of a safety-critical cyber-physical system," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3436–3446, 2017.

[52] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[53] T. de Bruin, J. Kober, K. Tuyls, and R. Babuška, "Integrating state representation learning into deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1394–1401, 2018.

[54] J. Chen, T. Shu, T. Li, and C. W. de Silva, "Deep reinforced learning tree for spatiotemporal monitoring with mobile robotic wireless sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[55] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[56] Y. Song, H. Lin, E. Kaufmann, P. Duerr, and D. Scaramuzza, "Autonomous overtaking in gran turismo sport using curriculum reinforcement learning," *arXiv preprint arXiv:2103.14666*, 2021.

[57] K. Shu, H. Yu, X. Chen, L. Chen, Q. Wang, L. Li, and D. Cao, "Autonomous driving at intersections: A critical-turning-point approach for left turns," *arXiv preprint arXiv:2003.02409*, 2020.

[58] K. Shu, H. Yu, X. Chen, S. Li, L. Chen, Q. Wang, L. Li, and D. Cao, "Autonomous driving at intersections: A behavior-oriented critical-turning-point approach for decision making," *IEEE/ASME Transactions on Mechatronics*, 2021.

[59] H. Kurniawati and V. Yadav, "An online pomdp solver for uncertainty planning in dynamic environment," in *Robotics Research*. Springer, 2016, pp. 611–629.

[60] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson, "Deep variational reinforcement learning for pomdps," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2117–2126.

[61] T. Liu, X. Mu, B. Huang, X. Tang, F. Zhao, X. Wang, and D. Cao, "Decision-making at unsignalized intersection for autonomous vehicles: Left-turn maneuver with deep reinforcement learning," *arXiv preprint arXiv:2008.06595*, 2020.

[62] 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*, 2022.

[63] C. Hu, L. Zhao, and G. Qu, "Event-triggered model predictive adaptive dynamic programming for road intersection path planning of unmanned ground vehicle," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 11 228–11 243, 2021.

[64] K. Wang, Y. Wang, L. Wang, H. Du, and K. Nam, "Distributed intersection conflict resolution for multiple vehicles considering longitudinal-lateral dynamics," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4166–4177, 2021.

[65] A. H. Hamouda, D. M. Mahfouz, C. M. Elias, and O. M. Shehata, "Multi-layer control architecture for unsignalized intersection management via nonlinear mpc and deep reinforcement learning," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 1990–1996.

[66] S. Noh, "Decision-making framework for autonomous driving at road intersections: Safeguarding against collision, overly conservative behavior, and violation vehicles," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3275–3286, 2018.

[67] M. A. Daoud, M. W. Mehrez, D. Rayside, and W. W. Melek, "Simultaneous feasible local planning and path-following control for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[68] J. Müller, J. Strohbeck, M. Herrmann, and M. Buchholz, "Motion planning for connected automated vehicles at occluded intersections with infrastructure sensors," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[69] S. Li, K. Shu, C. Chen, and D. Cao, "Planning and decision-making for connected autonomous vehicles at road intersections: A review," *Chinese Journal of Mechanical Engineering*, vol. 34, no. 1, pp. 1–18, 2021.

[70] M. Al-Sharman, D. Murdoch, D. Cao, C. Lv, Y. Zweiri, D. Rayside, and W. Melek, "A sensorless state estimation for a safety-oriented cyber-physical system in urban driving: deep learning approach," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 1, pp. 169–178, 2020.

[71] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[72] H. Liu, Z. Huang, and C. Lv, "Improved deep reinforcement learning with expert demonstrations for urban autonomous driving," *arXiv preprint arXiv:2102.09243*, 2021.

[73] M. Al-Sharman, R. Dempster, M. A. Daoud, M. Nasr, D. Rayside, and W. Melek, "Self-Learned Autonomous Driving at Unsignalized Intersections: A Hierarchical Reinforced Learning Approach for Feasible Decision-Making," 9 2022. [Online]. Available: https://www.techrxiv.org/articles/preprint/Self-Learned_Autonomous_Driving_at_Unsignalized_Intersections_A_Hierarchical_Reinforced_Learning_Approach_for_Feasible_Decision-Making/20770486

[74] R. Dempster, M. Al-Sharman, Y. Jain, J. Li, D. Rayside, and W. Melek, "Drg: A dynamic relation graph for unified prior-online environment modeling in urban autonomous driving," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8054–8060.

[75] C. Hu, S. Hudson, M. Ethier, M. Al-Sharman, D. Rayside, and W. Melek, "Sim-to-real domain adaptation for lane detection and classification in autonomous driving," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 457–463.

[76] R. Dempster, M. Al-Sharman, D. Rayside, and W. Melek, "Real-time unified trajectory planning and optimal control for urban autonomous driving under static and dynamic obstacle constraints," 2022. [Online]. Available: https://arxiv.org/abs/2209.09320

[77] M. S. Shirazi and B. T. Morris, "Looking at intersections: a survey of intersection monitoring, behavior and safety analysis of recent studies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 4–24, 2016.

[78] K. M. Dresner and P. Stone, "Sharing the road: Autonomous vehicles meet human drivers." in *Ijcai*, vol. 7, 2007, pp. 1263–1268.

[79] J. Alonso, V. Milanés, J. Pérez, E. Onieva, C. González, and T. De Pedro, "Autonomous vehicle control systems for safe crossroads," *Transportation research part C: emerging technologies*, vol. 19, no. 6, pp. 1095–1110, 2011.

[80] G. S. Aoude, B. D. Luders, K. K. Lee, D. S. Levine, and J. P. How, "Threat assessment design for driver assistance system at intersections," in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 1855–1862.

[81] K. Okamoto, K. Berntorp, and S. Di Cairano, "Driver intention-based vehicle threat assessment using random forests and particle filtering," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13 860–13 865, 2017.

[82] Y. Li, Y. Zheng, B. Morys, S. Pan, J. Wang, and K. Li, "Threat assessment techniques in intelligent vehicles: A comparative survey," *IEEE Intelligent Transportation Systems Magazine*, 2020.

[83] P. F. Orzechowski, A. Meyer, and M. Lauer, "Tackling occlusions & limited sensor range with set-based safety verification," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1729–1736.

[84] A. Iranitalab and A. Khattak, "Comparison of four statistical and machine learning methods for crash severity prediction," *Accident Analysis & Prevention*, vol. 108, pp. 27–36, 2017.

[85] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *arXiv preprint arXiv:1811.06711*, 2018.

[86] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[87] Y. Wang, J. Sun, H. He, and C. Sun, "Deterministic policy gradient with integral compensator for robust quadrotor control," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[88] K. Zhou, S. Song, A. Xue, K. You, and H. Wu, "Smart train operation algorithms based on expert knowledge and reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2020.

[89] S. S. Gu, T. Lillicrap, R. E. Turner, Z. Ghahramani, B. Schölkopf, and S. Levine, "Interpolated policy gradient: Merging on-policy and off-policy gradient estimation for deep reinforcement learning," in *Advances in neural information processing systems*, 2017, pp. 3846–3855.

[90] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[91] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[92] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Looking back on the actor-critic architecture," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[93] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE transactions on automatic control*, vol. 42, no. 5, pp. 674–690, 1997.

[94] T. D. Gillespie, *Fundamentals of vehicle dynamics*. Society of automotive engineers Warrendale, PA, 1992, vol. 400.

[95] J. Liu, Y. Luo, H. Xiong, T. Wang, H. Huang, and Z. Zhong, "An integrated approach to probabilistic vehicle trajectory prediction via driver characteristic and intention estimation," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3526–3532.

[96] Y. Wang, Z. Liu, Z. Zuo, Z. Li, L. Wang, and X. Luo, "Trajectory planning and safety assessment of autonomous vehicles based on motion prediction and model predictive control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8546–8556, 2019.

[97] S. Yang, W. Wang, C. Liu, and W. Deng, "Scene understanding in deep learning-based end-to-end controllers for autonomous vehicles," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 53–63, 2018.

[98] J. M. Yoo, Y. Jeong, and K. Yi, "Virtual target-based longitudinal motion planning of autonomous vehicles at urban intersections: Determining control inputs of acceleration with human driving characteristic-based constraints," *IEEE Vehicular Technology Magazine*, vol. 16, no. 3, pp. 38–46, 2021.

[99] C.-J. Hoel, K. Wolff, and L. Laine, "Tactical decision-making in autonomous driving by reinforcement learning with uncertainty estimation," *arXiv preprint arXiv:2004.10439*, 2020.

[100] Y. Liu, P. Zhao, D. Qin, G. Li, Z. Chen, and Y. Zhang, "Driving intention identification based on long short-term memory and a case study in shifting strategy optimization," *IEEE Access*, vol. 7, pp. 128 593–128 605, 2019.

[101] A. Trende, A. Unni, J. Rieger, and M. Fraenzle, "Modelling turning intention in unsignalized intersections with bayesian networks," in *International Conference on Human-Computer Interaction*. Springer, 2021, pp. 289–296.

[102] J. M. Scanlon, R. Sherony, and H. C. Gabler, "Predicting crash-relevant violations at stop sign–controlled intersections for the development of an intersection driver assistance system," *Traffic injury prevention*, vol. 17, no. sup1, pp. 59–65, 2016.

[103] Z. R. Doerzaph, "Development of a threat assessment algorithm for intersection collision avoidance systems," Ph.D. dissertation, Virginia Tech, 2007.

[104] C. Laugier, I. E. Paromtchik, M. Perrollaz, M. Yong, J.-D. Yoder, C. Tay, K. Mekhnacha, and A. Nègre, "Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety," *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 4, pp. 4–19, 2011.

[105] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Evaluating risk at road intersections by detecting conflicting intentions," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4841–4846.

[106] A. Zyner, S. Worrall, J. Ward, and E. Nebot, "Long short term memory for driver intent prediction," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1484–1489.

[107] D. J. Phillips, T. A. Wheeler, and M. J. Kochenderfer, "Generalizable intention prediction of human drivers at intersections," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1665–1670.

[108] A. Zyner, S. Worrall, and E. Nebot, "A recurrent neural network solution for predicting driver intention at unsignalized intersections," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1759–1764, 2018.

[109] Y. Jeong, S. Kim, and K. Yi, "Surround vehicle motion prediction using lstm-rnn for motion planning of autonomous vehicles at multi-lane turn intersections," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 1, pp. 2–14, 2020.

[110] A. Girma, S. Amsalu, A. Workineh, M. Khan, and A. Homaifar, "Deep learning with attention mechanism for predicting driver intention at intersection," *arXiv preprint arXiv:2006.05918*, 2020.

[111] A. Bender, J. R. Ward, S. Worrall, and E. M. Nebot, "Predicting driver intent from models of naturalistic driving," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 1609–1615.

[112] V. Gadepally, A. Krishnamurthy, and U. Ozguner, "A framework for estimating driver decisions near intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 637–646, 2013.

[113] N. Khairdoost, M. Shirpour, M. A. Bauer, and S. S. Beauchemin, "Real-time driver maneuver prediction using lstm," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 714–724, 2020.

[114] S. Beauchemin, M. Bauer, D. Laurendeau, T. Kowsari, J. Cho, M. Hunter, and O. McCarthy, "Roadlab: An in-vehicle laboratory for developing cognitive cars," in *Proc. 23rd Int. Conf. CAINE*, 2010.

[115] M. N. Azadani and A. Boukerche, "A novel multimodal vehicle path prediction method based on temporal convolutional networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[116] A. Sarkar, K. Czarnecki, M. Angus, C. Li, and S. Waslander, "Trajectory prediction of traffic agents at urban intersections through learned interactions," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–8.

[117] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decision-making under uncertainty for autonomous driving using continuous pomdps," in *17th international IEEE conference on intelligent transportation systems (ITSC)*. IEEE, 2014, pp. 392–399.

[118] V. Sezer, T. Bandyopadhyay, D. Rus, E. Frazzoli, and D. Hsu, "Towards autonomous navigation of unsignalized intersections under uncertainty of human driver intent," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3578–3585.

[119] D. Isele, A. Nakhaei, and K. Fujimura, "Safe reinforcement learning on autonomous vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–6.

[120] J. Bernhard, S. Pollok, and A. Knoll, "Addressing inherent uncertainty: Risk-sensitive behavior generation for automated driving using distributional reinforcement learning," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2148–2155.

[121] C.-J. Hoel, T. Tram, and J. Sjöberg, "Reinforcement learning with uncertainty estimation for tactical decision-making in intersections," *arXiv preprint arXiv:2006.09786*, 2020.

[122] I. Osband, J. Aslanides, and A. Cassirer, "Randomized prior functions for deep reinforcement learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 8617–8629.

[123] P. Zhu, X. Li, P. Poupart, and G. Miao, "On improving deep reinforcement learning for pomdps," *arXiv preprint arXiv:1704.07978*, 2017.

[124] Z.-W. Hong, S.-Y. Su, T.-Y. Shann, Y.-H. Chang, and C.-Y. Lee, "A deep policy inference q-network for multi-agent systems," *arXiv preprint arXiv:1712.07893*, 2017.

[125] W. Song, G. Xiong, and H. Chen, "Intention-aware autonomous driving decision-making in an uncontrolled intersection," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[126] M.-Y. Yu, R. Vasudevan, and M. Johnson-Roberson, "Occlusion-aware risk assessment for autonomous driving in urban environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2235–2241, 2019.

[127] ——, "Risk assessment and planning with bidirectional reachability for autonomous driving," *arXiv preprint arXiv:1909.08059*, 2019.

[128] X. Lin, J. Zhang, J. Shang, Y. Wang, H. Yu, and X. Zhang, "Decision making through occluded intersections for autonomous driving," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 2449–2455.

[129] D. Kamran, C. F. Lopez, M. Lauer, and C. Stiller, "Risk-aware high-level decisions for automated driving at occluded intersections with reinforcement learning," *arXiv preprint arXiv:2004.04450*, 2020.

[130] M. Naumann, H. Konigshof, M. Lauer, and C. Stiller, "Safe but not overcautious motion planning under occlusions and limited sensor range," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 140–145.

[131] Y. Jeong, K. Yi, and S. Park, "Svm based intention inference and motion planning at uncontrolled intersection," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 356–361, 2019.

[132] G. Li, S. Li, S. Li, and X. Qu, "Continuous decision-making for autonomous driving at intersections using deep deterministic policy gradient," *IET Intelligent Transport Systems*, 2021.

[133] X. Xiong, J. Wang, F. Zhang, and K. Li, "Combining deep reinforcement learning and safety based control for autonomous driving," *arXiv preprint arXiv:1612.00147*, 2016.

[134] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, G. Dulac-Arnold *et al.*, "Deep q-learning from demonstrations," *arXiv preprint arXiv:1704.03732*, 2017.

[135] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6292–6299.

[136] Z. Qiao, J. Schneider, and J. M. Dolan, "Behavior planning at urban intersections through hierarchical reinforcement learning," *arXiv preprint arXiv:2011.04697*, 2020.

[137] Z. Huang, J. Wu, and C. Lv, "Efficient deep reinforcement learning with imitative expert priors for autonomous driving," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[138] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of sumo-simulation of urban mobility," *International journal on advances in systems and measurements*, vol. 5, no. 3&4, 2012.

[139] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," *arXiv preprint arXiv:1711.03938*, 2017.

[140] F. Heidecker, J. Breitenstein, K. Rösch, J. Löhdefink, M. Bieshaar, C. Stiller, T. Fingscheidt, and B. Sick, "An application-driven conceptualization of corner cases for perception in highly automated driving," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 644–651.

[141] M. Pitropov, D. E. Garcia, J. Rebello, M. Smart, C. Wang, K. Czarnecki, and S. Waslander, "Canadian adverse driving conditions dataset," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 681–690, 2021.

[142] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *arXiv preprint arXiv:2002.00444*, 2020.

[143] X. Pan, Y. You, Z. Wang, and C. Lu, "Virtual to real reinforcement learning for autonomous driving," *arXiv preprint arXiv:1704.03952*, 2017.

[144] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3722–3731.

[145] C. Hu, S. Hudson, M. Ethier, M. Al-Sharman, D. Rayside, and W. Melek, "Sim-to-real domain adaptation for lane detection and classification in autonomous driving," *arXiv preprint arXiv:2202.07133*, 2022.

[146] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[147] A. Bewley, J. Rigley, Y. Liu, J. Hawke, R. Shen, V.-D. Lam, and A. Kendall, "Learning to drive from simulation without real world labels," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4818–4824.

[148] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 1–8.

[149] A. Amini, I. Gilitschenski, J. Phillips, J. Moseyko, R. Banerjee, S. Karaman, and D. Rus, "Learning robust control policies for end-to-end autonomous driving from data-driven simulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1143–1150, 2020.

[150] M. K. Al-Sharman, Y. Zweiri, M. A. K. Jaradat, R. Al-Husari, D. Gan, and L. D. Seneviratne, "Deep-learning-based neural network training for state estimation enhancement: application to attitude estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 1, pp. 24–34, 2019.

[151] C. Lv, Y. Liu, X. Hu, H. Guo, D. Cao, and F.-Y. Wang, "Simultaneous observation of hybrid states for cyber-physical systems: A case study of electric vehicle powertrain," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2357–2367, 2017.

[152] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing letters*, vol. 3, pp. 18–23, 2015.

[153] G. Xiong, F. Zhu, X. Liu, X. Dong, W. Huang, S. Chen, and K. Zhao, "Cyber-physical-social system in intelligent transportation," *IEEE/CAA Journal of Automatica Sinica*, vol. 2, no. 3, pp. 320–333, 2015.

[154] L. Li, X. Peng, F.-Y. Wang, D. Cao, and L. Li, "A situation-aware collision avoidance strategy for car-following," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 5, pp. 1012–1016, 2018.

[155] Y. Li, C. Lv, J. Zhang, Y. Zhang, and W. Ma, "High-precision modulation of a safety-critical cyber-physical system: Control synthesis and experimental validation," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 6, pp. 2599–2608, 2018.

[156] F.-Y. Wang, N.-N. Zheng, D. Cao, C. M. Martinez, L. Li, and T. Liu, "Parallel driving in cpss: A unified approach for transport automation and vehicle intelligence," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 577–587, 2017.

[157] T. Liu, H. Yu, H. Guo, Y. Qin, and Y. Zou, "Online energy management for multimode plug-in hybrid electric vehicles," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4352–4361, 2018.

[158] J. Tan, C. Xu, L. Li, F.-Y. Wang, D. Cao, and L. Li, "Guidance control for parallel parking tasks," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 301–306, 2019.

[159] L. Wang, Z. Zhan, X. Yang, Q. Wang, Y. Zhang, L. Zheng, and G. Guo, "Development of bp neural network pid controller and its application on autonomous emergency braking system," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1711–1716.

[160] C. Qiu, G. Wang, M. Meng, and Y. Shen, "A novel control strategy of regenerative braking system for electric vehicles under safety critical driving situations," *Energy*, vol. 149, pp. 329–340, 2018.

[161] N. Ding and X. Zhan, "Model-based recursive least square algorithm for estimation of brake pressure and road friction," in *Proceedings of the FISITA 2012 World Automotive Congress*. Springer, 2013, pp. 137–145.

[162] G. Jiang, X. Miao, Y. Wang, J. Chen, D. Li, L. Liu, and F. Muhammad, "Real-time estimation of the pressure in the wheel cylinder with a hydraulic control unit in the vehicle braking control system based on the extended kalman filter," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 231, no. 10, pp. 1340–1352, 2017.

[163] L. Li, J. Song, and Z.-q. Han, "Hydraulic model and inverse model for electronic stability program online control system," *Chinese Journal of Mechanical Engineering*, vol. 44, no. 2, p. 139, 2008.

[164] J. Zhang, C. Lv, J. Gou, and D. Kong, "Cooperative control of regenerative braking and hydraulic braking of an electrified passenger car," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 226, no. 10, pp. 1289–1302, 2012.

[165] K. O'Dea, "Anti-lock braking performance and hydraulic brake pressure estimation," SAE Technical Paper, Tech. Rep., 2005.

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

[167] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[168] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.

[169] P. Baldi and P. Sadowski, "The dropout learning algorithm," *Artificial intelligence*, vol. 210, pp. 78–122, 2014.

[170] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for lvcsr using rectified linear units and dropout," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 8609–8613.

[171] F. B. Naeini, A. M. AlAli, R. Al-Husari, A. Rigi, M. K. Al-Sharman, D. Makris, and Y. Zweiri, "A novel dynamic-vision-based approach for tactile sensing applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 5, pp. 1881–1893, 2019.

[172] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep model based domain adaptation for fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2296–2305, 2016.

[173] M. Xia, T. Li, L. Xu, L. Liu, and C. W. De Silva, "Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 101–110, 2017.

[174] J. Pan, Y. Zi, J. Chen, Z. Zhou, and B. Wang, "Liftingnet: A novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 6, pp. 4973–4982, 2017.

[175] H. Oh, J. H. Jung, B. C. Jeon, and B. D. Youn, "Scalable and unsupervised feature engineering using vibration-imaging and deep learning for rotor system diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3539–3549, 2017.

[176] L. Yao and Z. Ge, "Deep learning of semisupervised process data with hierarchical extreme learning machine and soft sensor application," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1490–1498, 2017.

[177] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[178] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[179] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[180] Y. Yuan, J. Zhang, Y. Li, and C. Li, "A novel regenerative electrohydraulic brake system: Development and hardware-in-loop tests," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 11 440–11 452, 2018.

[181] C. Lv, Y. Xing, C. Lu, Y. Liu, H. Guo, H. Gao, and D. Cao, "Hybrid-learning-based classification and quantitative inference of driver braking intensity of an electrified vehicle," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5718–5729, 2018.

[182] H. Bock and K. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems*," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984, 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2-6 July 1984. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017612059

[183] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[184] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[185] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 78. PMLR, 13–15 Nov 2017, pp. 1–16.

[186] P. Developers, "Parl," https://github.com/PaddlePaddle/PARL, 2021.

[187] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*. PMLR, 2018, pp. 1587–1596.

[188] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[189] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, 2015, pp. 1889–1897.

[190] Y. Ma, C. Sun, J. Chen, D. Cao, and L. Xiong, "Verification and validation methods for decision-making and planning of automated vehicles: A review," *IEEE Transactions on Intelligent Vehicles*, 2022.