Center for Advanced Multimodal Mobility Solutions and Education

Project ID: 2022 Project 03

ONLINE COOPERATIVE LANE-CHANGING MODEL OF CONNECTED AND AUTONOMOUS VEHICLES

Final Report

by

Wei Fan (ORCID ID: https://orcid.org/0000-0001-9815-710X)
Yang Zhao (ORCID ID: https://orcid.org/0000-0002-8204-8918)

Wei Fan, Ph.D., P.E.
Director, USDOT CAMMSE University Transportation Center
Professor, Department of Civil and Environmental Engineering
The University of North Carolina at Charlotte
EPIC Building, Room 3261, 9201 University City Blvd, Charlotte, NC 28223
Phone: 1-704-687-1222; Email: wfan7@uncc.edu

for

Center for Advanced Multimodal Mobility Solutions and Education
(CAMMSE @ UNC Charlotte)
The University of North Carolina at Charlotte
9201 University City Blvd
Charlotte, NC 28223

September 2022

ACKNOWLEDGEMENTS

This project was funded by the Center for Advanced Multimodal Mobility Solutions and Education (CAMMSE @ UNC Charlotte), one of the Tier I University Transportation Centers that were selected in this nationwide competition, by the Office of the Assistant Secretary for Research and Technology (OST-R), U.S. Department of Transportation (US DOT), under the FAST Act. The authors are also very grateful for all of the time and effort spent by DOT and industry professionals to provide project information that was critical for the successful completion of this study.

DISCLAIMER

Table of Contents

List of Figures

List of Tables

EXECUTIVE SUMMARY

Connected and automated vehicle (CAV) technologies are combination technologies of connected vehicle and automated vehicle. As widely known, CAVs can bring with them many benefits including improving safety, reducing emissions, and increasing mobility of the transportation system. As one of the hot studies that researchers pay attention to, LC (LC) safety control plays a significant role in Autonomous Vehicles' development. Thus, the LC behavior of the vehicle in connected environment becomes the focus of the current scholar's research.

Thanks to the advanced communication technologies, the LC behavior in AV shows a different characteristic from that in the traditional condition. The critical problems are dynamic trajectory planning, complex cooperation, and strict calculation requirements. As for LC maneuver, the state space and action space of leading vehicles, rear vehicles, and the current vehicles should be defined more carefully for platooning. The interaction with surrounding vehicles makes the LC behavior have a feature of cooperation. Motivated by this purpose, the decision-making and trajectory planning of the current vehicle and surrounding vehicles should be planned synchronously.

This study conducts several simulation-based experiments to develop an online LC decision-making model and trajectory planning model for AVs involving more than two vehicles. The LC decision model and optimal control algorithm are developed based on the risk awareness of human drivers, solved by the Priority Reply Deep Q Network (PRDQN). The trajectory planning model based on Inverse Reinforcement Learning (IRL) is constructed dynamically considering the time spatial variance during trajectory prediction and the time efficiency during safety control. To obtain valid results, various driving behavior parameters are calibrated to the real traffic conditions for human-driven vehicles. In particular, the calibration is conducted using Support Vector Machine (SVM) for LC trajectory parameters such as lateral movement, lateral acceleration, and steering angle. After the calibration process, the simulation is conducted on the straight freeway segment under mixed traffic environment including regular human-driven vehicles and AVs. Simulation results are discussed in detail. Overall, the result of this study gains a better understanding of AVs' LC in terms of time efficiency and safety control. It will play a significant role on transferring the CAVs technologies from the laboratory to the real-world market.

Chapter 1. Introduction

1.1 Problem Statement

Relieving traffic congestion, enhancing mobility, and reducing fuel emissions were perceived as the main benefits of connected and autonomous vehicles (CAVs) when they were first introduced. In the last decades, increasingly significant multidisciplinary efforts have been jointly made by the automotive industry, high-tech companies, public sectors, and research institutions around the world in this domain. However, most research efforts have been mainly focused on longitudinal control, such as car following models of Adaptive Cruise Control (ACC), and Cooperative ACC (CACC). In comparison, fewer contributions have been made towards lateral control maneuvers although lane changing behavior is extremely important and its relevant study becomes more and more critical particularly as a higher level of vehicle automation is enabled in the transportation system.

Lateral maneuver research is a challenging undertaking that optimizes vehicle and transportation system controls by considering individual vehicle kinetics and traffic flow system harmonization comprehensively. In general, the CAV LC (LC) system consists of four key modules: vehicle-to-vehicle (V2V) communication, localization, LC decision and planning, and vehicle control algorithm. Researchers have fully explored the advantages of several different vision capture and positioning technologies, such as light detection and ranging (LiDAR), radar, global navigation satellite system (GNSS), differential global position system (DGPS), and high-definition (HD) map construction. These technologies lay a solid foundation for LC model construction and optimal control algorithm development. In consideration of the cost and safety concerns associated with field tests, the LC model is mainly developed by using a typical simulation platform, such as SUMO or VISSIM at a microscopic level. Limited by traffic flow organization patterns of these simulation software platforms, the LC decision model and its optimal control in the real world may not be simulated and replicated as precisely as possible.

More precisely, studies on LC decision and planning should focus on when and where to perform LC maneuvers, which should be based on real-time information and decision models to help with their real-world implementation. In this regard, two main research gaps still exist: 1) Decision models may be made “jointly” in a collaborative manner rather than merely “simultaneously”; and 2) Lane changing decisions should be made in consideration of their impact on the traffic stream. In particular, the presence of the mixed traffic flow that consists of CAVs, autonomous vehicles (AVs), connected vehicles (CVs), and human vehicles (HVs), would make it more difficult to optimally decide the relevant LC behaviors, especially under many complex driving scenarios.

In addition, after making LC decisions, CAVs need to plan one or more reference trajectories that are appealing for a set of objectives and update the proposed trajectories in real time to avoid potential collisions until the LC process is completed. Due to the time delay in sensing, communication, computation, and execution, CAVs need to predict the possible behaviors of surrounding vehicles online to account for effective information transformation. The time-consuming control algorithms should be replaced by adaptive concurrent computation processes and logic.

This study will fill this gap by conducting several simulation-based experiments to develop a cooperative lane change decision-making model for CAVs involving more than two vehicles and different CAV penetration rates. The LC decision model and optimal control algorithm would be developed and simulated on different road segments, including the mainline, merging, and departure areas of freeways. This study will also aim to gain a better understanding of relevant computational efficiency in terms of time by an online computation process in order to help transfer the CAVs technologies from the laboratory to the real-world market.

1.2 Objectives

The main goal of this research is to construct an online cooperative LC decision and planning model. The objectives of this project are to 1) conduct a comprehensive literature review on cooperative LC decisions and planning of CAVs; 2) focus on the trajectory optimization model from the perspective of multi-objectives including safety, stability, and comfort of passengers, and improve the online algorithm's computational efficiency; 3) analyze the disturbance effect of LC maneuver on traffic flow characteristics; and 4) rebuild the LC decision models in a cooperative manner with more than two CAVs in two lanes.

1.3 Expected Contributions

To accomplish these objectives, several tasks have been undertaken. A comprehensive LC model based on the reinforcement learning-based optimization methodology has been developed to solve the decision-making and trajectory planning for CAVs. In particular, it should be mentioned that the developed reinforcement learning methodology is currently regarded as the most efficient learning-based model because of its low computation complexity and high efficiency. So, this control theory could be used to make optimal decisions for automated LC maneuvers, and the personal stylized trajectory planning from the perspective of behavior cloning.

An extensive review of the existing literature reveals that real-time control in dense traffic scenarios is so significant for the LC models. Several machine learning models have been proposed, however little success has been achieved to strike an ideal balance between global optimal solutions and computation complexity. The reinforcement learning method, especially the inverse reinforcement learning method, intends to find local optimal solutions when the rewards function converges. The decision-making models based on PRDQN and the trajectory planning models based on MEDIRL proposed in this research will bridge the research gap in that aspect.

1.4 Report Overview

The research will be structured as shown in Figure 1.1. In this chapter, the background and motivation of the study have been discussed, followed by the research objectives, and expected contributions.

Chapter 2 presents a comprehensive literature review on the state-of-the-art and state-of-the-practice of CAV technologies and various methodological approaches to study LC with or without CAVs. This chapter gives a clear picture of existing LC models with consideration of CAVs, possible modeling scenarios, traffic flow impact analysis at different penetration rates and simulation tools (PTV VISSIM, SUMO, CARLA) to evaluate the model's performance.

Chapter 3 describes the National Generation SIMulation (NGSIM) database that will be used as the data source. The data was collected through a network of synchronized digital video cameras. NGVIDEO, a customized software application developed for the NGSIM program, transcribed the vehicle trajectory data from the video. This vehicle trajectory data provided the precise location of each vehicle within the study area every one-tenth of a second, resulting in detailed lane positions and locations relative to other vehicles.

Chapter 4 presents critical problems of LC decision-making and trajectory planning based on the Markov Decision Process. Definitions of discrete state and action space of LC agent vehicles as well as the reward function are given here. To coordinate with the simulation via MPC, some redefinitions based on kinematic models are used to transmit vehicle trajectories from the previous machine learning output. Specifically, some considerations for smoothing discrete decisions in a time series are also illustrated in this chapter.

Chapter 5 discusses how to find the optimal policy to describe the ground truth under the assumption of behavior cloning. The mechanism of two typical reinforcement learning models, PRDQN and Maximum Entropy Deep Inverse RL (MEDIRL), are illustrated. In this regard, the general optimization framework is formulated. The objective function is defined to minimize the risk level or collision of LC agent vehicles and best mimic the human driving data from Chapter 3.

Chapter 6 describes the fundamental settings of the automated LCs control process in CARLA and the dense traffic simulation scenarios.

Chapter 7 presents results of the four simulation scenarios in detail. The collision number, statistical analysis of gap distance, acceleration, and jerk angle of each scenario are analyzed with different combinations of Human-driven vehicles (HVs) and AVs, so that the effects of success LC rate of AVs could be quantified.

Chapter 8 concludes the report with a summary of the simulation results. Directions for future work will also be provided.

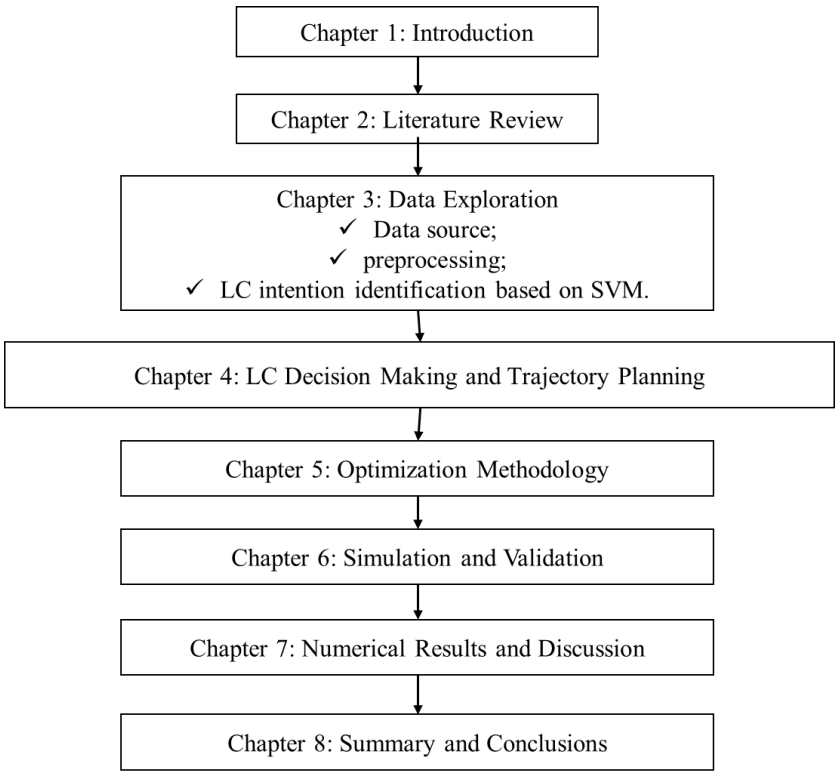

Figure 1.1 Research structure

Chapter 2. Literature Review

2.1 Introduction

The intelligence of automobile products has become the main trend of development, and related technologies are actively researched and developed by automobile manufacturers and scientific research institutes, which can drive a significant change. The major research efforts on Connected and Autonomous Vehicles (CAV) technologies include cooperative collision avoidance, platooning, cooperative cruise control, and lane changing assistance (LC). LC technology is arguably the most complex and critical to enabling higher levels of vehicle automation. It pertains to opportunistic or mandated transfer of a host CAV from the current lane to an adjacent lane. In general, the CAVs' LC system consists of four key modules: vehicle-to-vehicle (V2V) communication, localization, decision and planning, and vehicle control algorithms. With the help of V2V communication, the position, speed, and acceleration of the surrounding vehicles can be acquired faster and more accurately.

Because of the negative impact on traffic safety and linkage to macroscopic traffic flow characteristics, compared with car-following (CF), the LC model has not received much attention until recently. With the realization of LC's significant impacts on traffic safety and traffic congestion, efforts to model it have rapidly increased over the last decade. Briefly, LC is often distinguished as either *discretionary* or *mandatory*; this is because each of these LC types generally involves different decision-making processes and has different impacts on surrounding traffic. The primary purpose of a discretionary LC is to gain a speed advantage or a better driving environment, whereas the primary motivation of a mandatory LC is to reach the planned destination. Meanwhile, the modeling efforts in the literature roughly fall into two components: modeling the LC decision-making process (i.e., how a driver reaches the LC decision-making is not specified) and trajectory planning process.

The following sections are organized as follows. Section 2.2 presents the background CAV technologies, such as definitions, taxonomies, impacts, and prospects. Section 2.3 summarizes four typical LC models from previous studies and their related work, including the primary theory, functions, model calibration process, scenarios, contributions, and limitations. The impact analysis and performance measurement are presented in Section 2.4. Section 2.5 illustrates the most used microscopic simulation platforms, specifically the PTV VISSIM, SUMO, and CARLA. Finally, a summary of the chapter is given in Section 2.6.

2.2 Background of LC Process

2.2.1 LC Process

With the help of the information from V2V communication and localization, the LC decision module can predict surrounding vehicles' movements and decide when and where to perform the LC operation. Then, the LC planning model can generate one or more reference trajectories from the current position to the target position, concerning safety and comfort.

To accomplish the task, a trajectory-tracking controller should be developed to track the predefined trajectories according to the CAV state and road information. The differences between the actual trajectory and reference trajectory, in terms of errors on longitudinal and

lateral locations, and the CAV's heading angle will be corrected. The vehicle control algorithms can then be converted into an input to the actuators. The desired speed and yaw rate are realized by controlling the torque and the steering wheel, respectively. These four key modules cooperate continuously to complete the task. Besides the addressed individual vehicle control technologies, cooperative maneuvers from multiple vehicles in traffic have been shown to impact overall traffic flow performance significantly.

2.2.2 Scope

In this report, the literature review on CAV's LC model will be illustrated into two-parts: (1) four classic category models based on different theories and some typical constructed historical models, which play a significant role in this field, and (2) the summarization of their main contributions, highlights, and limitations, which would lay a solid foundation for processing research work. Additionally, some popular microscopic simulation software will be summarized to form the basis of further research. Considering the availability of LC model testing in the academic research, the previous studies on field test will not be discussed.

2.3 Classic LC Model of Connected and Autonomous Vehicles

2.3.1 Rule-Based Models

2.3.1.1. Gipps LC Model

Gipps et al. (1986) was among the first research teams to introduce a structure of LC for drivers who faced conflicting goals. Gipps' model described the LC decisions and the execution of lane changes on freeways and urban streets as the result of three factors: LC possibility, the necessity for changing lanes, and LC desirability. More specifically, factors that impact LC in Gipps' model included the possibility of changing lanes without an unacceptable risk of collision, the locations of permanent obstructions, the presence of heavy vehicles, the presence of special purpose lanes (e.g., transit lanes), the driver's intended turning movement, and the possibility of gaining a speed advantage. In this model, a driver's behavior fell into three zones, which do not depend on the driver's behavior patterns over time. Thus, this model was deterministic, which implies that each of the rules is evaluated sequentially according to its importance.

After Gipps' pioneering work, many efforts either extend or improve his LC modeling framework. Trade-offs among considerations, such as the variation among different drivers and the inconsistency in a driver's behavior over time, were ignored.

Yang and Koutsopoulos (1996) developed and implemented a LC model in the microscopic traffic simulator, MITSIM. They classified LC as either mandatory or discretionary, and modeled LC in a sequential four steps: decision to consider a LC, choice of the target lane, searching for an acceptable gap, and executing the change. Although the rule-based modeling framework of that was similar to Gipps', a distinct feature of their model was that, instead of treating LCD as a deterministic process, LC probability was introduced to make the model more realistic.

2.3.1.2. CORSIM Model

Halati et al. (1997) developed a LC model that was implemented in CORiodr SIMulation (CORSIM), in which lane changes were classified as mandatory lane changing (MLC), discretionary lane changing (DLC), and random lane changing (RLC). MLC occurred when drivers merge onto a freeway or move to the target lane to make an intended turn or avoid obstructions (e.g., lane blockage and lane drop) in a lane. DLC was applied when lane changes are required for speed advantage. For instance, a driver may want to pass a slow-moving vehicle by changing to the left lane. RLC was applied when there is no apparent reason. RLC may or may not result in an advantage for the subject vehicle over its current position. In CORSIM, a certain percentage (the default value was 1%) of drivers were randomly selected to perform RLC. In this model, motivation, advantage, and urgency were considered as the three major factors behind a LC decision. The motivation to change lanes depended upon either the lead vehicle speed or the lead headway threshold. The advantage factor captured the benefits of driving in the target lane. The urgency of lane changing depended upon the number of lanes to change and the distance required to execute a complete LC maneuver.

2.3.1.3. FRESIM Model

FRESIM Model was firstly proposed in 2004. In the FRESIM DLC procedure, LC benefits were referred to as "Advantage." Advantage was modeled through either the "lead factor" or "putative factor." The disadvantage of staying in the current lane was represented by the lead factor. On the other hand, the putative factor represented the benefits of executing lane changes. Theoretically, a subject vehicle driver could select any one of the adjacent lanes (left/right) as the target lane for performing lane changes. Thus, the advantage was calculated for both adjacent lanes through the putative factor. The overall advantage for DLC was represented by the difference between the putative factor and the lead factor. It was then compared with a threshold value of 0.4. If the overall advantage was greater than the threshold value, a lane change occurred. So far, only the FRESIM DLC model has been discussed. The RLC and MLC were also incorporated in FRESIM. LC algorithms used in the FREeway SIMulator (FRESIM) and NETwork SIMulator (NETSIM) were similar. NETSIM was firstly proposed in the same year and improved in 2007. The only difference lied in measuring gaps between the subject vehicle and the lead/lag vehicles in the target lane. NETSIM measured the gaps in terms of time differences, and the gaps in FRESIM were a function of time headways and speed differences.

2.3.1.4. ARTEMIS Model

ARTEMiS, which is an abbreviation for Analysis of Road Traffic and Evaluation by Micro-Simulation, was a microscopic traffic simulation model developed by Hidas et al. (2002). Previously named Simulation of Intelligent TRAnsport Systems (SITRAS), this model described LC maneuvers based upon the courtesy of the lag vehicle in the destination lane. In this model, a lane change was triggered by required downstream turning movements, lane drops, lane blockages, lane use restrictions, speed advantages, or queue advantages. MLC occurred in the case of downstream turning movements, lane

drops, and lane blockages, and DLC happened in the early and middle-distance zones. The boundaries of different zones were defined in the same way as Gipps' model.

Hidas et al. (2005) modeled each vehicle as a driver-vehicle object (DVO), using an autonomous agent technique to describe drivers' interactions involved in a complex decision-making process. DVOs can act as giving way, slowing down, or not giving way, based on road congestion conditions, individual driver characteristics, and the perception of a DVO in terms of whether another DVO was trying to move into its lane or not. According to this model, LC reasons were evaluated, and the results were classified as "essential," "desirable," or "unnecessary," based on which a target lane was chosen.

In ARTEMiS, gap acceptance model selection depended on LC modes. Two LC modes were proposed according to traffic conditions and the necessity of changing lanes: normal lane changing and courtesy/forced lane changing. A normal lane change occurred when a sufficient gap is available in the target lane. This LC mode was based on the Hidas' car-following model and can be expressed as: 1) acceptable deceleration (or acceleration) was required for the subject vehicle to follow the lead vehicle in the target lane, and 2) acceptable deceleration was required for the lag vehicle in the target lane, so that the subject vehicle can safely serve as its lead vehicle.

2.3.1.5. Cellular-Automata Based LCD Model

Cellular automata (CA) was historically proposed in the 1940s and popularized in the 1980s to accurately reproduce macroscopic behavior of a complex system using minimal microscopic descriptions. A typical CA model constituted four key components: the physical environment, the cells' states, the cells' neighborhoods, and local transition rules, as denoted in (1).

$$CA=(\zeta,\Sigma,N,\delta) \quad (1)$$

Where ζ is the physical environment represented by the discrete lattice; Σ represents the set of possible states; N denotes the neighboring cells; and δ is the local transition rules, which are commonly given by a rule table.

CA models were frequently applied in various fields, including traffic flow modeling. Several notable traffic CA (TCA) models were developed for reproducing CF and LC behaviors, such as single-cell models, multi-cell models, deterministic models (e.g., Wolfram's rule 184 (Wolfram, 1983), deterministic Fukui–Ishibashi TCA (Fukui and Ishibashi, 1996)); Stochastic models (e.g., Nagel–Schreckenberg TCA, STCA with cruise control, Stochastic Fukui–Ishibashi TCA (Nagel and Schreckenberg, 1992; Nagel, 1995); STCA with cruise control (Nagel and Paczuski, 1995), Stochastic Fukui–Ishibashi TCA (Fukui and Ishibashi, 1996)); Slow-to-start models (e.g., Takayasu–Takayasu TCA (Takayasu and Takayasu, 1993)); Velocity-dependent randomization TCA (e.g., Barlovic´ et al., 1998; Barlovic´, 2003). To demonstrate the setup of a typical TCA, a single-cell CA model using Wolfram's rule 184 (which is defined later) for a single lane road was presented here. Other models can be seen in Chowdhury et al. (2000), Knospe et al. (2004), Nagel (1996), Schadschneider (2000, 2002), Schreckenberg et al. (2001), and Maerivoet and Moor (2005).

Like traditional traffic flow theories, the longitudinal movements of individual vehicles in TCA were also governed by CF. In fact, TCA was closely connected to traditional traffic flow theories. For example, a TCA can be derived from Gipps' CF model (Gipps, 1981); and Daganzo (2004) proved two TCA models' equivalence to the kinematic wave model with a triangular fundamental diagram.

2.3.1.6. Game Theory Model

The game theory model was based on the give way behavior in a merging situation when a traffic conflict arises between through and merging vehicles, in which they intend to influence each other. Kita et al. (1999) modeled this situation based upon the game theory and specified the game type, the number of players, and the repetition of games. They also considered the cooperative nature of the game.

In this research work, it was assumed that the games are independent, and the strategies of each player (i.e., the payoff matrices) are known by the other player and noncooperative because both players had information on each other. These two players played two different strategies: "merge" and "pass" for the merging vehicle and "give way" and "do not give way" for the through vehicle. If the merging vehicle and the through vehicles were denoted by player 1 (X1) and player 2 (X2), respectively, the pure strategy of X1m was:

$$m = \{1\text{:}\, merge, 2\text{:}\, pass\}.$$

And the strategy of X2n was:

$$n = \{I\text{:}\, giveway, \text{II: do not giveway}\}$$

Whether a merging car merges or a through car gives way depended on the given situation with a certain probability. Both players used mixed strategies for this type of situation. For a mixed strategy game, a bi-matrix provided at least one equilibrium solutions.

Kita et al. (1993) modeled on-ramp merging behavior using a discrete choice model, and the probability of giving way was estimated based on this game theory model. In Kita's model, drivers compared the utilities of the current lane and the target lanes (left/right) and chose the target lane with a higher utility. In this case, the utilities perceived by the drivers captured the payoff of the players.

For congested traffic conditions, Pei and Xu (2006) developed another LC model based on game theory for two types of LC maneuvers. Traffic information and experience were the basis of their model to describe lane changing maneuvers. In their model, cooperative and forced lane changes were also defined. The values of time and safety were the main factors affecting driver behavior. When drivers were safe situations, they would execute a LC maneuver. The game theory model was largely limited to describing the merging–give way behavior in freeway merging areas and cannot be easily extended to other LC maneuvers.

Table 2-1 Rule based LC models

No.	Author	Year	Rule based	Data source	Optimization	Simulator	Scenarios
1	Gipps	1986	Gipps Model	Australian	Gain a speed advantage	-	Urban street
2	Yang and Koutsopoulos	1996	Gipps Model	-	Gap distance	MITSIM	Freeway
3	Halati	1997	Gipps Model	-	-	-	Mandatory LC; discretionary LC
4	Kita	1999	Maximum-likelihood	-	Discrete choice model (LC probability)	-	Merging
5	Hidas	2002, 2005	Maximum-likelihood	Video data	Speed flow curve	-	On ramp; weaving area
6	FHWA Off. Oper. Res	2004, 2007	FREeway SIMulator (FRESIM);	Field test	Gap acceptance	FRESIM	Freeway
7	Pei and. Xu	2006	Maximum likelihood	-	Nonlinear programming	-	Congestion

2.3.2 Discrete-Choice-Based Models

2.3.2.1. Ahmed's model

Ahmed et al. (1999) adopted utility theory to model the decision process of LC. The proposed LCD structure consisted of four latent (i.e., unobserved) levels of decision hierarchy in driving characteristics across the driving population and explanatory variables that affect driver behaviors. Ahmed extended the mandatory LC model to specifically accommodate heavily congested traffic, where forced merging behaviors frequently occur because of lacking normally acceptable gaps. Two levels were involved in this decision process: intention of merging into the target lane, and perception on the establishment of a mutual understanding on right of way. This two-level decision process was evaluated at every discrete time point and the forced merging begins.

Ahmed subsequently implemented his model in MIcroscopic Traffic SIMulator (MITSIM). It was developed primarily to assess advanced traffic management systems and advanced traveler information systems at the operational level. Although his LC model was unable to capture the tradeoffs between MLC and DLC decision processes, it accurately described the differences between drivers' MLC, DLC, and FM decisions. For instance, in MITSIM, drivers were unable to overtake when mandatory considerations are active. Similar to the Gipps model, the existence of an MLC was determined based upon the distance of the subject vehicle to the downstream exit ramp. In addition, a dummy variable was introduced to capture the differences in acceptable gap values between a passenger car and a heavy vehicle when the heavy vehicle was the subject. Although this very coarse and simplistic method accounted for the differences in operational characteristics of these two vehicle types, the aforementioned models incorporated a rigid separation between MLC and DLC, which was unrealistic in real-life driving.

2.3.2.2. Toledo et al' s Model

Actually, the boundaries between mandatory LC and discretionary LC exist sometimes since the discretionary LC usually would be considered first before mandatory LC. And the previous mentioned model failed to capture the trade-offs between them because a rigid separation was assumed among them. Moreover, the mandatory LC situation was not always perceived by the driver (except for special cases like on-ramp merging traffic). Hence, the conditions that trigger a mandatory LC had not been estimated in the models above.

To overcome this problem, Toledo et al. (2002) proposed an integrated LC model where mandatory and discretionary conditions were joined together in a single utility model. The model captured the trade-offs between the utility of being in the correct lane (i.e., the mandatory LC consideration) and that of the speed advantage offers by a faster lane (i.e., the discretionary LC consideration). The model also considered a driver specific random term that represents unobservable characteristics of the driver and correlations between observations of the same driver over time. Parameters of the model were jointly estimated using vehicle trajectories collected from I-395 Southbound, Arlington, Virginia, US. And results showed the importance of incorporating trade-offs between the mandatory and the discretionary LC into the model.

Although driver characteristics (e.g., level of aggressiveness, alertness) naturally had significant impact on various aspects of lane change decision making process, they were missing from most of the existing LCD models. To explicitly incorporate the effect of driver characteristics, Sun and Elefteriadou (2011) conducted a focus group study to identify and understand drivers' concerns and responses under various LC scenarios. From the focus group study, driver types, and reasons and main factors for each driver type in lane changing decision-making processes were revealed and linked. To observe drivers' actions under various LC scenarios, and to obtain field-measured values for the important factors identified in the focus group study, field data were collected using instrumented vehicles.

2.3.2.3. Markov Process Based LCD Models

LC had also been modeled as a Markov process. The first Markov-based LC model was perhaps proposed in Worrall et al. (1970), where a stochastic LC model is developed as a homogeneous Markov chain and calibrated using data collected on a section of 6-lane freeway in Chicago.

In a broader context of treating human as a device with a large number of unobservable internal mental states, Pentland and Liu (1999) modeled the driving behavior using a Markov dynamic model. LC experiments using driving simulator were used to demonstrate the soundness of the proposed modeling framework. LC was broken down into a chain of states: (1) a preparatory centering the car in the initial lane; (2) looking around to make sure the target lane is clear; (3) steering to initiate LC; (4) the change itself; (5) steering to terminate the change; and (6) a final re-centering of the car in the target lane. Results supported the view that human actions are best described as a sequence of control steps rather than as a sequence of raw positions and velocities. In the case of driving, this meant that the action is defined by the pattern of acceleration and heading.

Sheu and Ritchie (2001) modeled the mandatory LC induced by incidents using the Markov process. All state variables in the stochastic system followed homogeneous Gaussian–Markov processes. Unlike the previous studies in which stable traffic conditions were often assumed to justify the use of a time-invariant transitional LC probability, a noise term which follows a Gaussian process was introduced to accommodate time-varying traffic conditions that are caused by incidents.

The models described above in this section aimed to reproduce LC frequency but could not explain the decision process: why or why not LC occurs. Thus, they were not suitable for microscopic simulations. This limitation was overcome in Toledo and Katz (2009) by integrating a hidden Markov model (HMM) with the utility theory-based modeling.

2.3.2.4. Hazard-based (Survival) LCD Models

Hamdar (2009) criticized previous LC models for neither sufficiently nor explicitly considering stochasticity and possibly unsafe character of the cognitive processes (e.g., perception, judgment, and execution) followed by drivers. Thus, a hazard-based duration model was proposed. Unlike rule-based LC models, the hazard-based duration model treated driver behaviors as a multiple duration process: Free flow, CF, or LC. Three

parametric hazard functions were adopted by Hamdar (2009): the increasing monotonic dependence; non-monotonic dependence; and the third one was based on an increasing positive correlation between duration and hazard before a given time, followed by a constant hazard value. The proportional hazard form was employed to accommodate effects of exogenous factors (e.g., headways, speed, speed difference, etc.).

It was expanded to accommodate the fact that multiple types of events may end a CF, LC, or free-flow process. Two strategies were discussed: (1) the utility-based strategy: each potential event that ends a particular state was considered as an alternative with a given utility. Then the appropriate exit strategy was determined according to these utilities; and (2) the hazard-based strategy: instead of using utilities to determine the appropriate exit strategy, the exit strategy with the associated highest hazard was selected. It was not clear which strategy was employed in their study. The NGSIM vehicular trajectories (Alexiadis et al., 2004) were used to calibrate and validate their model.

Table 2-2 Discrete choice based LC model studies

No.	Author	Year	Rule based	Data source	Optimization	Simulator	Scenarios
1	Ahmed	1996	Discrete choice	Vehicle trajectories	Gap distance	-	Freeway
2	Ahmed	1999	Discrete choice	Vehicle trajectories from I-93, Boston, US	Gap distance	SUMO	Freeway
3	Toledo	2003	Discrete choice	Vehicle trajectories from I-395 Southbound, Arlington, Virginia, US	Driver behaviors (steering angle, acceleration, deceleration)	VISSIM SUMO	Freeway
4	Sun and Elefteriadou	2011, 2012	Markov Process	Field data from instrumented vehicles	Driver characteristics	-	Congested arterial
5	Hidas	2002, 2005	Markov Process	Video	Speed-flow curves	VISSIM	Urban & freeway
6	Worrall	1970	Markov Process	A section of 6-lane freeway in Chicago	Probability of LC	SUMO	Freeway
7	Pentland and Liu	1999	Markov Process	Driving simulator	Positions and velocities (Acceleration, heading)	Driving simulator	Merging area
8	Sheu and Ritchie	2001,2012	Markov Process	Loop detector data	Driver behaviors (steering angle, acceleration, deceleration)	VISSIM	Incident
9	Toledo and Katz	2009	Markov Process	Vehicle trajectories from I-395 southbound in Arlington, Virginia, US	Positive and significant state-dependency coefficient	VISSIM	Incident
10	Hamdar	2009	Hazard	NGSIM vehicular trajectories	Perception, judgment, and execution	VISSIM	Intersection

Table 2-3 Intensive based models

No.	Author	Year	Model Based	Data source	Optimization model	Simulator	Scenarios
1	Kesting	2007		European countries	MOBIL		Congestion
2	Schakel	2012		Segment of A20 Freeway near Rotterdam, Netherlands	Speeds of the free-flow condition		Freeway

2.3.3 Incentive Based Models

2.3.3.1. Minimizing Overall Braking Included by Lane Changes (MOBIL)

Kesting et al. proposed a novel logic for simplifying and modeling LCD, which is MOBIL. The MOBIL LC model was based on two criteria: incentive and safety. The incentive criterion measured the attractiveness of a given lane based on its utility, and the safety criterion measured the risk associated with lane changing (i.e., acceleration).

The MOBIL rules were applied for simulation of multilane traffic in the intelligent driver model (IDM). In IDM, two types of passing rules were considered for lane changes: symmetric and asymmetric. The symmetric passing rules were based on safety and incentive criteria. They were applied when changing to the right lane is not strictly forbidden. When the deceleration of the follow vehicle in the target lane was equal to the IDM braking deceleration, the safety criterion was satisfied. For a lane change to happen, the deceleration of the follow vehicle should also not exceed a certain b_{safe} limits, as shown below. Thus,

$$a'(F') > -b_{safe} \tag{2}$$

Where $a'(F')$ is the deceleration of the immediate follower in the target lane; b_{safe} denotes the safe limit. For countries without such asymmetric lane usage rules like US, the desirability rule was:

$$\tilde{a}_c - a_c + p(\tilde{a}_n - a_n + \tilde{a}_0 - a_0) > \Delta a_{th} \tag{3}$$

Where a_c and $\tilde{a}_c$ denote the lane changer's acceleration before and after the lane change, respectively; other variables are the same as previously defined.

As indicated in the lane changing rules above, MOBIL considered the advantage or disadvantage of the followers via a politeness factor. By adjusting this parameter, the motivations for lane changing can be varied from purely egoistic to more cooperative behavior, e.g., increasing the combined accelerations of the lane changer and affected neighbors.

Using accelerations in MOBIL had two main advantages: (1) the lane change decision-making process was dramatically simplified, which leads to the parsimoniousness of MOBIL; and (2) accelerations could be readily calculated with an underlying microscopic longitudinal traffic model, which enables MOBIL to be easily integrated with a typical CF model. However, the logic of MOBIL had yet to be empirically justified, and MOBIL itself had yet to be calibrated and validated.

2.3.3.2. LC Model with Relaxation and Synchronization (LMRS)

Schakel et al. (2012) proposed a LMRS, based on drivers' desire to change lanes. The desire was a combination of the route, speed, and keep-right incentives. A tradeoff was considered within the combination of incentives, with the route incentive being dominant. The following equation was a sample combination of incentives representing the desire to change from lane i to lane j:

$$d^{ij} = d_r^{ij} + \theta_v^{ij} * (d_s^{ij} + d_b^{ij}) \quad (4)$$

Where, d^{ij} is the combined desire to change lane from i to j; d_r^{ij} denotes the desire to follow a route; d_s^{ij} represents desire to gain speed; d_b^{ij} represents desire to keep right; and θ_v^{ij} is the voluntary (discretionary) incentives.

The total desire determined drivers' LC behaviors. The range of meaningful desire was from -1 to 1. Negative values represented that a lane change is not desired, and positive values meant that the driver wants to change lane. Depending upon the desire value, Schakel et al. (2012) further classified lane changes as free lane changing (FLC), synchronized lane changing (SLC), and cooperative lane changing (CLC). Thus,

$$0 < d_{free} < d_{sync} < d_{coop} < 1 \quad (5)$$

The gap acceptance module in this model was similar to MOBIL. In addition, this model considered an applicable headway for gap acceptance. A gap was accepted if the accelerations of the subject vehicle and the new follower were larger than a safe deceleration threshold. According to this model, large decelerations and short headways could be accepted for a large desire, and the relaxation of headway values was exponential with relaxation time. The subject vehicle driver would synchronize her/his speed, if the LC desire was above the synchronization threshold (d_{sync}). She/he would synchronize the speed with the target lane speed by applying a maximum deceleration, which is both comfortable and safe. A gap can be created if an adjacent leader LC desire was above cooperation threshold.

2.3.4 Artificial Intelligence Models

2.3.4.1. Fuzzy-logic-based Models

Fuzzy-logic-based models considered the uncertainty of lane changing maneuvers, and the natural or subjective perception of real variables was also considered. The unique nature of fuzzy logic models was that they can translate nonlinear systems into IF-THEN rules. The following was a typical IF-THEN LC rule:

IF: $(vehicle\ i\ is\ eligible\ for\ using\ the\ left\ lane)\ and$

$(the\ gap\ between\ vehicle\ i\ and\ the\ leader\ in\ the\ left\ lane\ is\ large) and$

$(the\ gap\ between\ vehicle\ i\ and\ the\ follower\ in\ the\ left\ lane\ is\ large) and$

$(the\ speed\ in\ the\ current\ lane\ is\ low)\ and$

$(the\ speed\ in\ the\ left\ lane\ is\ high)$

$THEN$: $(vehilce\ i\ changes\ to\ the\ left\ lane)$.

Using fuzzy logic was often reported to be capable of better mimicking a driver's actual decision process because fuzzy logic was well equipped to handle human's cognitive and

perceptional uncertainties frequently encountered in real-world LC processes. Technically speaking, all the models discussed above can be fuzzified. For example, Yeldana et al. (2012) proposed a TCA model based on fuzzy logic. Several fuzzy logic-based LCD models were developed (McDonald et al., 1997; Brackstone et al., 1998; Das et al., 1999; Wu et al., 2000; Moridpour et al., 2009).

2.3.4.2. Artificial Neural Network (ANN) Models

ANN models process information using functional architecture and mathematical models that are similar to the neuron structure of the human brain. These models learn human behaviors from training and can demonstrate those human behaviors in a new situation. In recent years, neural networks have also been used for modeling driver behavior in the transportation field. For instance, Hunt and Lyons (1994) predicted drivers' LC decisions using neural networks on dual carriageways. Neural network models were completely data driven and required supervised training by field-collected traffic data before they could be used to predict driving behavior. Their dependence on the availability of field-collected traffic data was the main disadvantage of neural network models, although previous results showed that they can accurately predict LC behavior.

Dumbuya et al. (2009) developed neural driver agents (NDAs) for modeling LC maneuvers. A multilayer NDA model was designed and implemented. A back-propagation training algorithm was used to train the NDA model, which takes inputs such as current direction of the vehicle, current speed, distance from the vehicle, preferred speed, and current lane. The output of the model included new direction and new speed. This NDA model learnt LC behavior from known situations using data collected from the Transport Research Laboratory (TRL) driving simulator. The authors then used the fitted NDA model to predict driver behavior for unseen situations.

Codevilla et al. (2018) proposed a conditional imitation learning method to generate driving policies as a chauffeur to handle sensorimotor coordination. The continuous response to navigational commands helped make decisions to avoid obstacles in the experiments.

Xu et al. (2017) used the long short-term memory architecture to encode the instantaneous monocular camera observations and previous vehicle states, and their network was then trained to imitate drivers' realistic behaviors based on a large-scale video dataset.

These methods were in an end-to-end manner, benefiting from the most significant advantage of supervised learning that the relationship between sensor inputs and model outputs can be directly mapped by using the developed network (Xu et al., 2017).

2.3.4.3. Reinforcement Learning based Models

With the rapid development of vehicle-to-vehicle and vehicle-to-infrastructure technology, LC models in a connected environment (i.e., vehicles are connected and assisted by the system where the surrounding traffic data are shared) instead of a traditional environment (i.e., a conventional environment where there is no assistance system and drivers have to make the decision based on their experience) have been

proposed frequently in the literature. Among the learning-based methods, reinforcement learning (RL) has obtained a lot of attention in recent years and has achieved superhuman performance in this field.

Mnih et al., (2015); Hasselt et al., (2015); Schaul et al., (2016); Duan et al., (2021) made a major breakthrough in deep reinforcement learning (DRL) in recent years. They started to apply DRL to address the driving decision-making problems in autonomous driving (Shin et al., 2019; Long et al., 2018; Li et al., 2019; Ye et al., 2019; Zhu et al., 2020). DRL-based methods can greatly decrease the heavy reliance on a large amount of data because they do not need labeled driving data for training (Zhu et al., 2018; Moghadam and Elkaim, 2019; Hoel et al., 2020).

Kahn et al. (2017) proposed an uncertainty-aware reinforcement learning algorithm to learn obstacle avoidance strategies by using uncertainty to estimate driving risk.

He et al. (2018) implemented a novel RL approach to enable dynamic orchestration of networking, caching, and computing resources while they propose a multi-agent actor-critic method to learn better resource management strategies to meet vehicle services.

Wang et al. (2018) proposed a reinforcement learning-based approach to train the agent to learn an automated lane change behavior so that it could intelligently make a lane change for collision avoidance under unforeseen scenarios.

Long et al. (2018) presented a DRL-based system-level scheme for multi-agents to plan their own collision-free actions without observing other agents' states and intents.

Moghadam and Elkaim (2019) introduced DRL into a hierarchical architecture to make a sequential tactical decision (e.g., lane change) for AVs to avoid collisions, and then the tactical decision was converted to low-level actions for vehicle control. Unlike supervised learning methods, DRL-based methods can compensate for the high cost of data collection in dangerous scenarios by training models in virtual simulation environments with affordable trial-and-errors.

Bouton et al. (2019) proposed a probabilistic guarantee-based reinforcement learning strategy for autonomous driving at intersections.

Mokhtari and Wagner (2021) developed a risk-based decision-making framework that integrates risk-based path planning with reinforcement learning-based control for safe driving. Although risk assessment has already been considered in some of the DRL-based decision-making methods, the examination of risk assessment consideration for autonomous driving in lane change scenarios still needed to be further investigated (Chen et al.)

Table 2-4 Artificial intelligence LC model studies

No.	Author	Year	Model based	Optimization	Simulator	Scenarios
1	McDonal	1997	Fuzzy logic	Mimic drivers' decision-making process on LC	-	Freeway
2	Brackstone	1998	Fuzzy logic	Mimic drivers' decision-making process on LC	VISSIM	Freeway
3	Das	1999	Fuzzy logic	Mimic drivers' decision-making process on LC	VISSIM	Freeway
4	Wu	2000	Fuzzy logic	Mimic drivers' decision-making process on LC	-	Merging
5	Moridpour	2009	Fuzzy logic	Mimic drivers' decision-making process on LC	-	Freeway
6	Ross	2010	Fuzzy logic	Mimic drivers' decision-making process on LC	-	Freeway
7	Yeldana	2012	Fuzzy logic	Mimic drivers' decision-making process on LC	-	Freeway
8	H.Yang	1992	Deep learning	Predict drivers' driving behavior in the next time step	VISSIM	Merging
9	Hunt	1994	Deep learning	Predict driver behavior	VISSIM	Urban
10	Dumbuya	2007	Deep learning	Cooperative LC process	Neuro Solutions software	Urban
11	Li	2018	Deep learning	Minimize the errors between target state and the predicted state	SUMO	Merging
12	Suh	2018	Deep learning	Cooperative LC process	SUMO	Freeway
13	Codevilla	2018	Deep learning	Conditional imitation learning method to generate driving policies	VISSIM	Merging
14	Hasselt	2015	Reinforcement learning	Driving decision making for LC	VISSIM	Static objectives
15	He	2018	Reinforcement learning	Driving decision making for LC	CARLA	Freeway
16	Moghadam and Elkaim	2019	Reinforcement learning	Compensate for the high cost of data collection in dangerous scenarios	CARLA	Dangerous area
17	Mokhtari and Wagner	2021	Reinforcement learning	Risk based path planning	CARLA	Freeway

2.4 Impacts on Network Performance

The impacts of autonomous vehicles could be studied by the changes in terms of the demand and behavior side, the supply of mobility services, and the network and facility operational performance. Unfortunately, the potential changes in the supply of transportation and mobility at the urban scale are difficult to predict and characterize for the purpose of developing specific planning tools and forecasting the demand for these services over time.

Fagnant and Kockelman (2015) conducted some research related to autonomous capabilities and argued that it would preclude individual vehicle ownership altogether in favor of shared mobility fleets.

Mahmassani (2016) adopted a broader portfolio of services, possibly in conjunction with third parties, the disappearance of conventional fixed-route, fixed-schedule bus service in the most low-density communities, supplanted by driverless, and personalized service at low density. The shared hybrid forms at medium densities and a greater focus on frequent rapid service along the dedicated right of way (rail and/or bus rapid transit) in higher density travel corridors were considered.

2.4.1 Impacts on Traffic Flow

The simplest way to understand the potential impact of CAVs on traffic flow is to go back to the basics of traffic science, namely to the interrelation between spacing, density, speed, and flow. It is doubtable that the traffic flow of the CAV environment is the same as the HDV, not mentioned to the mixed traffic flow.

In practical traffic situations, beyond the deployment status of the connectivity protocols and factory settings for acceptable following distances, at least three factors mitigate these increases: (1) weaving and lane changing losses in multilane situations; (2) mixed traffic implications under most deployment scenarios; and (3) flow stability considerations, particularly at high speed in heterogeneous environments.

Talebpour, Mahmassani, and Hamdar (2015) and Talebpour, Mahmassani, and Bustamante (2016) developed a game-theoretic LC model that captures the dynamic interactions between drivers during discretionary and mandatory LC maneuvers. And a game structure to model the behavior when drivers were not aware of the nature of the LC maneuver is introduced (i.e., mandatory vs. discretionary).

2.4.2 Performance Measurement

Talebpour and Mahmassani (2016) estimated the CAV simulation results of traffic flow regarding the stability and throughput of mixed traffic streams with varying compositions of autonomous and/or connected vehicles along with regular vehicles. The following part will illustrate these measurements from the above two.

2.4.2.1. Stability Analysis

Talebpour and Mahmassani (2016) first examined the stability question by extending the analytical approach developed by Ward (2009) and applied it in turn to systems with varying

degrees of connected vehicles in one case, then autonomous vehicles in the other. The analytical derivation was applied to Hamdar et al.'s (2018) car following model for regular vehicles.

Treiber, Kesting, and Helbing (2007) proposed the simulation-based investigation of string stability by adapting the methodology. The occurrence of different stability regimes (stable, oscillatory, and collision) was investigated for different platoon sizes, reaction times, and market penetration rates of connected and autonomous vehicles.

Treiber, Kesting, and Helbing's (2007) categorized stable, oscillatory, and collision regimes. The collision regime was identified that if perturbations lead to a crash at some point in the simulation. Finally, the oscillatory regime was identified if neither of these cases is recognized.

2.4.2.2. Throughput Analysis

Throughput effects under different market penetration rates of CAV are examined on a hypothetical one-lane highway with an on-ramp located in the middle of the segment. Talebpour and Mahmassani (2016) conducted that at high autonomous vehicle market shares, and the throughput was nearly doubled while the scatter was virtually eliminated, with no flow breakdown over the range of demand loading considered.

2.5 Simulation Platforms of LC Model

CAVs are a big part of the automotive industry's overall growth trend that may be utilized to improve transportation safety, expand mobility options, lower expenses, and provide new job possibilities. Thus, a complete examination of connected and autonomous driving is required before the large-scale implementation, which may be done affordably and efficiently using a reliable simulation platform. Current traffic simulators ease the operation of CAVs by offering incremental enhancements to traditional traffic flow modeling approaches, which cannot replicate the features of real-world connected and autonomous vehicles.

Generally, a microscopic traffic simulator consists of three major components: (1) a transportation network to define road topology at a network level; (2) a traffic demand generator to create traffic flow running in the predefined traffic network; and (3) a car-following and LC models to regulate individual vehicle driving behavior. In microscopic traffic models, vehicles are represented as separate agents, whose motion is governed by specific rules. Those agents may be in interaction, which also has an impact on their behaviors. Some typical simulators include PTV VISSIM, SUMO, and CARLA.

2.5.1 PTV VISSIM

VISSIM has been a widely used tool for simulating connected and automated vehicles. It provides different interfaces that allow adapting internal driving parameters: car-following model, lane change behavior, operating speed, and acceleration. VISSIM models multi-modal realistic transportation operations and creates the best conditions for testing different traffic scenarios before their realization. It has various features, such as traffic flow modeling, traffic light control, vehicle queue length analysis, pedestrian simulation, and script-based modeling. Script-based modeling was one of the VISSIM's features that are very

useful in the development of traffic control algorithms in the research studies of P. Gora et al., (2020) and E. Joelianto et al., (2019).

2.5.2 SUMO

SUMO allows modeling of intermodal traffic systems including road vehicles, public transit, and pedestrians. Included with SUMO is a wealth of supporting tools that handle tasks such as route finding, visualization, network import, and emission estimation. SUMO can be enhanced with custom models and provides various APIs to remotely control the simulation via a general traffic control interface (TraCI), which could make it possible to bi-directionally couple traffic simulators with other software.

2.5.3 CARLA

A lot of previous projects described how to use this software to simulate the microscopic LC behavior, such as A. Dosovitskiy et al., (2017). Besides, the literature review paper by F. Rosique et al., (2019) also mentioned the great achievement of this software in the CAV study.

2.6 Summary

A decent number of studies have been conducted on CAV lane changing from different disciplines (e.g., electrical engineering, computer science, transportation engineering, and mechanical engineering) with different focuses (e.g., LC model, impact analysis, and simulation platforms) in the last decades. Yet the existing studies are relatively segregated by disciplines or perspectives.

In summary, there is a clear need to develop a comprehensive model that captures the (mandatory or discretionary) LC decision-making process and its consequent impact on surrounding traffic. In developing a new LC model, the multi-level evaluation strategy should be generally preferred: at the macroscopic level, outputs of the model should be consistent with typical traffic flow characteristics; at the microscopic level, lane changing decisions need to be matched with observations with a reasonably low prediction error rate, and trajectories of the vehicles involved in LC should be close to actual trajectories in our future work.

Chapter 3. Data Exploration

3.1 Introduction

This chapter describes the National Generation SIMulation (NGSIM) database that is used as the data source for this project. The data was collected through a network of synchronized digital video cameras. NGVIDEO, a customized software application developed for the NGSIM program that transcribed the vehicle trajectory data from the video. This vehicle trajectory data provided the precise location of each vehicle within the study area every one-tenth of a second, resulting in detailed lane positions and locations relative to other vehicles. So, this kind of ground truth data could be used to simulate human driving behaviors.

The data source and preprocess work will be illustrated in this chapter. The technical details of LC trajectory data extraction will be described in the following parts. To identify and analyze the LC trajectory more specifically, the Support Vector Machine (SVM) classification model is applied to further analyze LC driving behavior.

3.2 Data Source

The dataset used to estimate and validate the model was recorded in Los Angeles, California from the plane of NGSIM proposed by the FHWA. The I-80 data file was eastward vehicle trajectories captured by 7 cameras synchronously on 13th, April 2005. The data was transferred every tenth of a second. The coverage area of the camera is about 500 meters, composed of six main lanes and one ramp. The supporting information contains the precise position, speed, and acceleration. In the past, it was used for driving behavior analysis, traffic parameters calibration, car-following and lane changing trajectory prediction.

The specific labels of this dataset are vehicle number, types, geometric features, lateral positions, vertical positions, lanes, speed, acceleration, and front-rear cars. The data description is shown in Table 3-1. Since every vehicle that appeared in the monitoring section owns one unique ID, its trajectory and other surrounding vehicles could be traced.

Table 3-1 Description of raw data

Name	Description	Units
Vehicle ID	According to the order entering the monitoring section; Identification by the license plate number	-
Frame ID	The time series is numbered from the beginning of detection	-
Total frames	Total number of vehicles in this frame	-
Total time	Calculated at 0 hours, 1970.1.1	ms
Partial x axis	The front center of the vehicle is X deviated from the starting point of the road in the monitoring area	ft
Partial y axis	The front center of the vehicle is Y deviated from the starting point of the road in the study area	ft
Overall x axis	The partial X relative to the CA State Plane Ⅲ in NAD83	ft
Overall y axis	The partial Y relative to the CA State Plane Ⅲ in NAD83	ft
Length of vehicle	Geometric length of vehicles	ft
Width of vehicle	Geometric width of vehicles	ft
Vehicle type	1-motor, 2-car, 3-trunk	-
Vehicle speed	Vehicle speed at current moment	ft/s
Vehicle acceleration	Vehicle acceleration at current moment	ft/s

Lane ID	lane at current moment	-
Front vehicles	Front vehicles ID	-
Rear vehicles	Rear vehicles ID	-
Space headway	The distance between the vehicle and the front vehicle	ft
Headway	The time between the vehicle and the front vehicle	s

3.3 Data Preprocessing

3.3.1 Data Filtering

3.3.2.1. Vehicle Type Identification

The NGSIM data set divides vehicles into three types: motor, car, and others (truck, bus), as shown in Table 3-2. The proportion of cars on the I-80 expressway is about 76.06% which plays the dominant role among all vehicle types during the detection period. Therefore, the vehicle type was selected as "2", which is the car corresponding trajectories, to gain a reasonable LC trajectories sample size.

Table 3-2 Summary of vehicle types in I-80 section

Vehicle type	I-80 road segment	
	Number of vehicles (PCU)	Ratio (%)
1-motor	25	8.80%
2-car	216	76.06%
3-track, bus	43	15.14%

3.3.2.2. LC Scenarios Identification

From Figure 3.1, the observed section of I-80 is composed of 7 lanes, in which lane 1 to lane 5 are the main lanes, and lane 6 and lane 7 are merging (diverging) lane. For that, the most frequent mandatory LC appears in lane 6 and 7. Typically, relative low speed and frequent lateral movement are the two most significant characteristics of these traffic flows. This is out of the scope of this research. The LC trajectories on lane 1-5 are taken and will be studied.

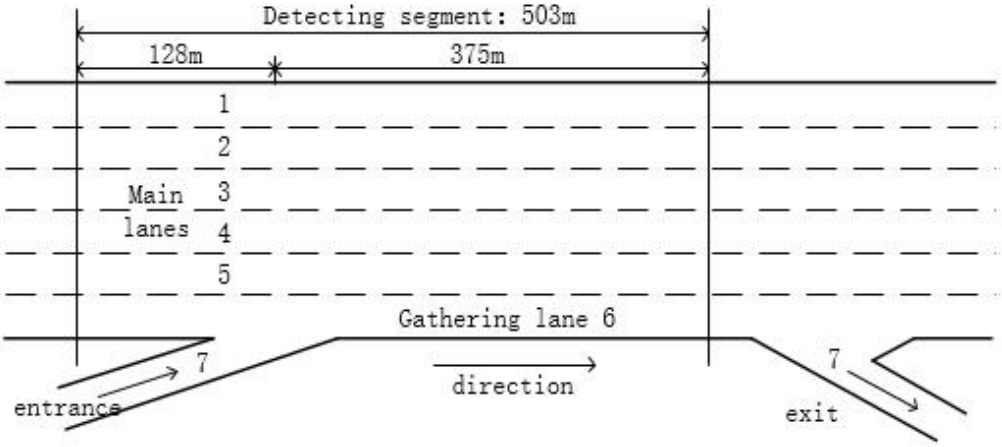

Figure 3.1 Geometric Composition of I-80 Highway

3.3.2.2. LC Trajectory Extraction

To improve the identification accuracy, some constraints have been assigned in this study:

(1) The LC that occurred at the merging and diverging lane is not studied.

(2) The LC that occurred at the boundary of the video detection segment is not studied.

(3) The lateral movement of vehicles is constrained by the following equation.

$$\begin{cases} \frac{x_1+x_2+w}{2} - c < x_s < \frac{x_1+x_2-w}{2} \\ \frac{x_1+x_2+w}{2} < x_F < \frac{x_1+x_2-w}{2} + c \end{cases} \quad (1)$$

Where, x_s、x_F is the lateral position at the beginning of LC; x_1、x_2 is the lateral position of vehicles changing lane ID at the front and rear moments; $(x_1 + x_2)/2$ is the central position of lane; w is the vehicle width; and c is the lane width, assuming it is 3.75m in average.

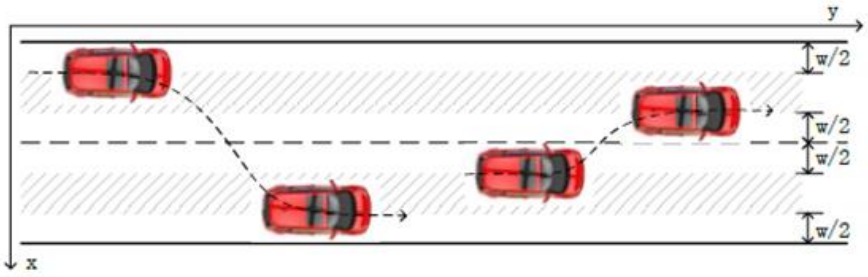

Figure 3.2 Lateral Constraints of Vehicle Lane Changing

The start and end point of LC are limited in the shadow area in Figure 3.2. Based on these assumptions and constraints, a total number of 210 vehicle trajectories are extracted on I-80 for further analysis in this project.

3.3.2 Data Filtering Summary

Based on these 210 single complete LC trajectories, the cooperative LC trajectories are further extracted by the surrounding vehicle IDs which have an interactive relationship with them. Figure 3.3 shows some representative statistical results, and the vehicle number of 0 indicates that there is no vehicle in this interactive location.

NO.	M	N1	N2	N3	N4
1	1773	1764	1780	1798	1781
2	1810	1735	1801	1825	0
3	1905	1895	0	1900	1910
4	2088	0	2061	2138	2173
5	3113	3110	3094	0	3127
6	3323	3315	3315	0	0
7	3355	0	0	1	44

Figure 3.3 Statistical results of interaction relationship between lane changing vehicles

3.3.3 Trajectory data smoothing

When the raw data was filtered, some noise appeared frequently in the vehicle speed and acceleration distribution. Since vehicle speed and acceleration are differential derived

parameters of longitudinal positions, abnormal unstable fluctuations will occur in the time-varying curves of them, which cannot be directly applied to the study of LC driving behavior and related models. The theory of NGSIM dealing with speed and acceleration is shown as:

$$\begin{cases} v(t) = \frac{x(t)-x(t-1)}{time\ step} \\ a(t) = \frac{v(t)-v(t-1)}{time\ step} \end{cases} \quad (2)$$

Where, x(t) is the longitudinal position at t; v(t) and a(t) are the differential speed and acceleration at the same time respectively; and the time step is the time interval of each frame, which is 0.1s.

According to the calculation method adapted by NGSIM, the detection error of longitudinal position at each moment would be amplified as ten times when calculating the vehicle speed, and hundreds of times when calculating vehicle acceleration.

Besides the calculation error, some individual perception can also be observed in the distribution of vehicle speed and acceleration. As for absolute value of acceleration, most of them are 3~3.5m/s^2, which is abnormal from the reality that drivers always accelerate or brake continuously, especially during the peak hour under dense traffic scenarios. These result in the velocity and acceleration noise when measured.

To eliminate these noises, the Symmetric Exponential Moving Average (SEMA) method is adapted to deal with all trajectories before a deep data mining. The theory of SEMA can be expressed by:

$$\begin{cases} x_{\alpha}^{'}(t) = \frac{\sum_{k=i-D}^{i+D} x_{\alpha}(t_k) e^{-|i-k|/\Delta}}{\sum_{k=i-D}^{i+D} e^{-|i-k|/\Delta}} \\ D = \max\{3\Delta, t-1, N_m - t\} \end{cases} \quad (3)$$

Where, $x'_{\alpha}(t)$ represents the approximate position of a certain vehicle α at t, $x_{\alpha}(t_k)$ denotes the original position of this vehicle at t_k, N_m represents the number of trajectory data, D is the smooth window width when considering boundary data. Δ is the smooth window width when considering intermediate data, $\Delta = \frac{\mathrm{T}}{d_t} = 10T$, T is the smooth time, which have not been determined by the investigators. Thiemann et al. (2008) recommended that the time of 4 seconds can rapidly decrease the acceleration variance at very small smooth time. Since the vehicle acceleration and speed are deduced step by step, the amplification error of longitudinal position is much less in speed than acceleration. Simultaneously, it takes T = 0.5s and T=1s respectively as the smooth time for position and speed through comparing different parameters. Based on the testing result, T=0.5, 1, 4s have been taken as the smooth time to deal with the position, speed, and acceleration of vehicle trajectories finally. Figure 3.4 provides an example of the smoothing.

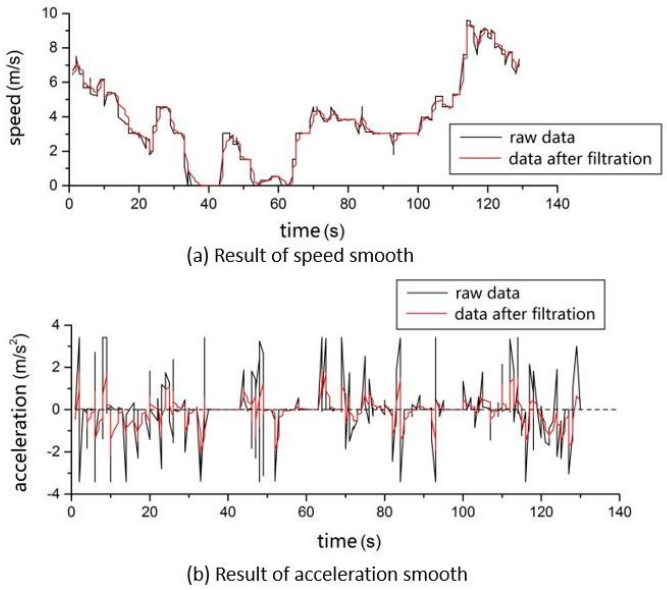

Figure 3.4 The smoothing results of vehicle trajectory data with ID = 1773

3.4 LC Intention Identification Based on SVM

3.4.1 Mechanism of SVM

SVM is a binary classification model, which aims to separate all samples by a hyper-plane limited by the principle of interval maximization. Hence, the classification process can be transferred into a convex quadratic programming problem, which helps obtain stronger generalization ability and a global optimal resolution at same time. There are three types of SVM, liner separation SVM, approximate liner separation SVM and nonlinear separation SVM. Since numeric complex factors attributes to the lane change intention model, this paper focuses more on the nonlinear separation SVM and its kernel function construction and soft separation maximization mechanism.

When dealing with non-linear classification problem, it usually maps the training set into a high dimensional space that can be separated. Suppose that the feature vectors after mapping is φ(x), and the raw training set is x. Thus, the hyper-plane can be demonstrated as:

$$f(x) = \omega \cdot \varphi(x) + b \tag{4}$$

Importing the slack vector ξ_i to soften the separation rules, it meets the following optimized requirements:

$$\begin{gathered} \max_{\omega,\ b,\xi_i} \frac{1}{2}\|\omega\|^2 + c\sum_{i=1}^{N}\xi_i \\ \text{s.t. } y_i(\omega \cdot \varphi(x_i) + b) \geq 1 - \xi_i \\ \xi_i \geq 0, i = 1,2,\dots,N \end{gathered} \tag{5}$$

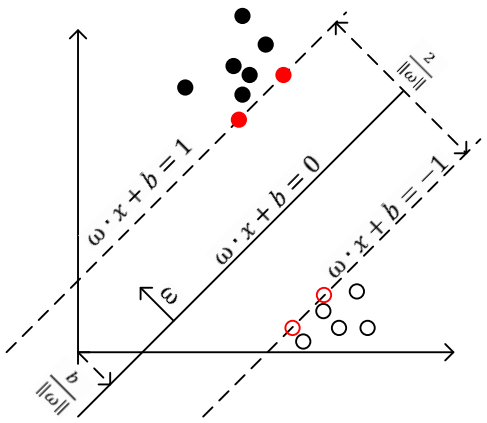

Figure 3.5 The optimal hyperplane

Where, $C > 0$ is a punishment parameter, which can be used to adjust the slacking degree. A bigger C means the more serious punishments on classification.

In order to simplify the solving process, the convex quadratic can be transferred into a dual problem by Lagrangian multiplier, which can be expressed as:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{N} \alpha_I - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\varphi(x_i) \cdot \varphi(x_j)) \\ &\text{s.t.} \sum_{i=1}^{N} \alpha_i y_i = 0, 0 \le \alpha_i \le C, i = 1, 2, \ldots, \end{aligned} \tag{6}$$

It is difficult to solve this dual equation directly because of inner product. Hence, based on functional theory, a kernel function that meets the Mercer theorem is induced to calculate the corresponding point product at high dimensional space. This kernel function can be expressed as:

$$\mathrm{k}(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle = \varphi(x_i) \cdot \varphi(x_j) \tag{7}$$

Thus, the equation 2.4 can be rewritten as:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{N} \alpha_I - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ &\text{s.t.} \sum_{i=1}^{N} \alpha_i y_i = 0, 0 \le \alpha_i \le C, i = 1, 2, \ldots, N \end{aligned} \tag{8}$$

After solving, it is:

$$\mathrm{f(x)} = \omega \cdot \varphi(\mathrm{x}) + \mathrm{b} = \sum_{i=1}^{N} \alpha_i y_i k(x_i, x_j) + b \tag{9}$$

3.4.2 Flowchart of SVM

The most critical problem of LC intention recognition is to figure out whether the vehicle is prepared to take an obvious lateral movement. The SVM model is used here to identify this problem. The flowchart of its training process is shown in Figure 3.6. Firstly, the feature vectors that can represent LC intention are selected. The feature dimensionality is conducted by the Principal Component Analysis (PCA). Then, the Grid Search Method (GSM) is used to find the optimal parameters for kernel function. The SVM classification model of is established using the Matlab LibSVM library. The Accuracy of the model is verified by the

accuracy value and the Accuracy Under Curve (AUC) value of Receiver Operating Characteristic (ROC) curve.

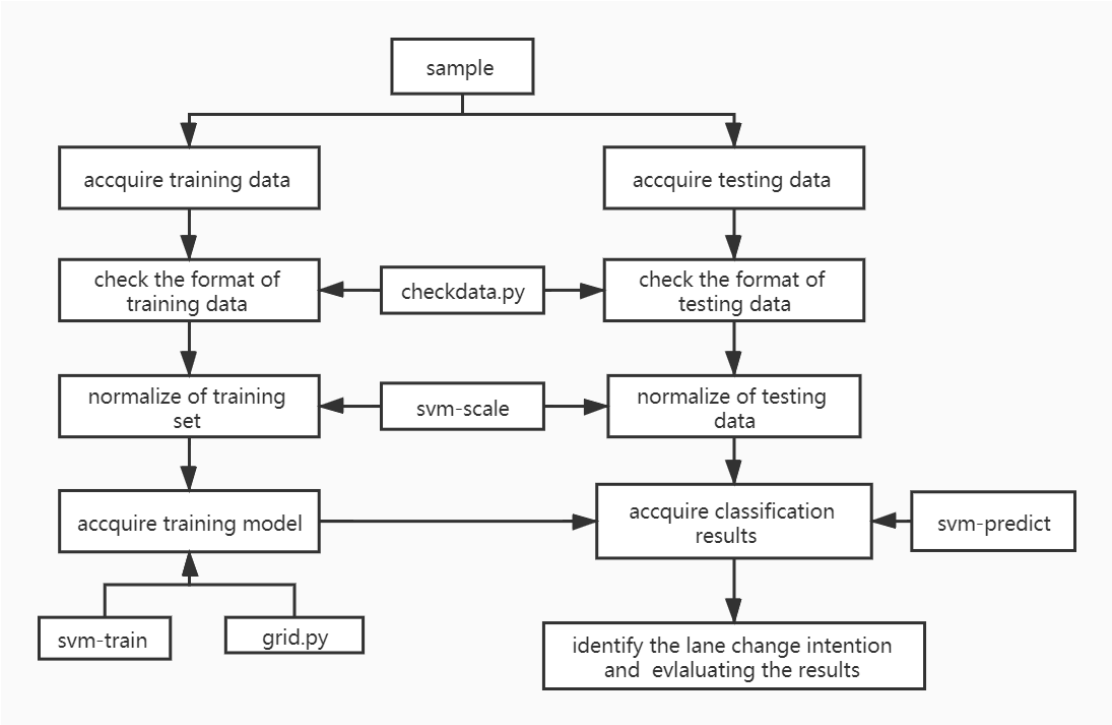

Figure 3.6 The flowchart of SVM classification

The LibSVM is a kind of open-source package of Matlab, which could be easily applied and modified. The four main py. files of LibSVM are subset.py, easy.py, grid.py, and checkdata.py. Three execution files of Windows that are set here to support LibSVM running are svm-scale, svm-train, svm-predict.

As recommended by Nagatani (1993) 0.2 m/s of lateral movement is preliminarily labeled as the beginning timestamp of LC intention. The training sample is extracted from the previous 5s of to the following 5s of this timestamp. All trajectories are labeled into positive, negative, and 0. Positive represents the vehicle that takes LC, negative represents the vehicle that takes car-following, and 0 represents others.

90% of the data is randomly selected as training sets respectively, and the remaining 10% is used as test data to verify the model identification. After the data filtering, a total number of 4,730 waypoints are extracted, including 1,130 for LC and 3,600 for car-following. The proportion of positive and negative samples in the sample is about 1:3, so the distribution proportion of the original data set is reasonable.

3.4.3 Training Details of SVM

3.4.3.1. Feature Engineering

When identifying whether the driver has LC intention, attention should be paid not only to the driving state of the vehicle itself and the difference among the surrounding vehicles, but also to the difference between the driving state of the vehicle in front of and behind the target lane. This study measures driver maneuver by the numeric method,

considering velocity, acceleration, steering angle, relative velocity, and relative gap distance. More specific labels used for SVM training are shown at Table 3-3.

Table 3-3 Selection of feature vectors

Feature ID	Feature	Units	Description
1	v_x	m/s	longitudinal speed of current vehicle
2	v_y	m/s	lateral speed of current vehicle
3	a_x	m/s^2	longitudinal acceleration of current vehicle
4	a_y	m/s^2	lateral acceleration of current vehicle
5	θ	deg	steering wheel angle of current vehicle
6	Δv_1	m/s	speed differences between the front vehicle and the current vehicle
7	Δv_2	m/s	speed differences between the target vehicle and the current vehicle
8	Δv_3	m/s	speed differences between the rear vehicle and the current vehicle
9	Δd_1	m	distance between the front vehicle and the current vehicle
10	Δd_2	m	distance between the target vehicle and the current vehicle
11	Δd_3	m	distance between the rear vehicle and the current vehicle

Redundant information that comes from various vectors could mislead the machine learning to obtain very fuzzy mapping results as well as low efficiency. To solve this problem, some measurements would be taken to optimize the features to readjust the vector space. In this report, the PCA method is used to reduce the vector's dimensions by deleting the interference among them. The steps of PCA are:

Step1: Calculating the covariance matrix and acquiring its feature vectors.

Step2: Ranking all feature vectors in descending order.

Step3: Acquiring the feature vectors and unifying them, then mapping the principal vectors by multiplying the transferred feature vectors matrix and data matrix.

Step4: Repeating step 3 until all data are mapped with the principal component.

After the PCA process, a feature more than 1 is selected as the principal component. The contribution degree of each feature is shown in Table 3-4. Since the cumulative contribution rate of the top 5 components is more than 87.808%, they are shown in Table 3-5 to input SVM for training.

Table 3-4 Component accumulation contribution rate of PCA

Feature ID	Feature	Contribution rate	Accumulative contribution rate
1	2.322	27.112	27.112
2	1.847	21.787	48.899
3	1.221	15.097	63.996
4	1.068	12.71	76.706
5	1.001	11.102	87.808
6	0.867	4.88	92.688
7	0.757	2.883	95.571
8	0.693	2.301	97.872
9	0.597	1.124	98.996
10	0.342	0.608	99.604
11	0.286	0.396	100

Table 3-5 Component score coefficient matrix of PCA

Feature Vector	Principle component				
	1	2	3	4	5
v_x	0.284	-0.103	0.254	-0.039	0.057
v_y	0.029	-0.084	0.269	0.49	0.689
a_x	0.072	0.485	0.009	-0.001	-0.001
a_y	-0.005	-0.284	-0.037	-0.251	0.05
θ	0.07	0.446	0.067	0.029	0.069
Δv_1	0.204	-0.043	-0.276	0.457	0.017
Δv_2	0.31	-0.001	-0.396	-0.154	0.096
Δv_3	-0.335	0.045	0.188	-0.034	-0.034
Δd_1	0.142	0.003	0.563	-0.11	-0.013
Δd_2	0.253	-0.028	0.253	-0.339	-0.126
Δd_3	0.074	-0.068	0.164	0.52	-0.698

3.4.3.2. Parameters Optimization

Among the Linear kernel, Polynomial kernel, Sigmoid kernel, and Radial Basis Function (RBF) of SVM, the last one is used in this paper because of its fewer hyperparameters and small sample dimensions. c and γ are the two dominant parameters for RBF. To ensure the drivers' intention classification accuracy, the GSM is used to search for the most optimized set by comparing all possible combinations.

Parameters for the superior high diagram are shown in Figure 3.7, and the $\log_2 c$ and $\log_2 g$ are the corresponding parameters c and γ to 2 at the bottom of the numerical value. Each parameter optimization steps by 0.5. According to the Matlab output optimization results and the comparison of the figure below, when the horizontal and vertical coordinate values are 3.5 and 4 respectively, the highest cross-validation accuracy is 95.2891%, when $c = 11.3137$, $\gamma = 16$.

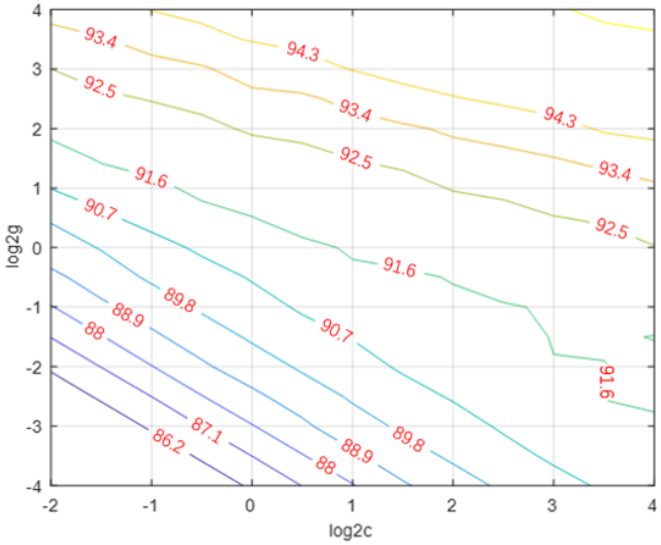

Figure 3.7 Contour plot of parameter optimization

3.5 Summary

The field detecting data from NGSIM of the I-80 highway in Los Angeles, California is used to capture the vehicles' trajectory features. A comprehensive introduction of NGSIM I-80 data has been presented in the preceding section. After the extraction of LC trajectory data, a total of 4,730 data samples are extracted, including 1,130 samples for lane changing and 3,600 for lane keeping. The SVM classification algorithm is used to identify the driving behavior of car-following and LC. The perfect value of 95.2891% cross-validation accuracy when $c = 11.3137$, $\gamma = 16$ from the GSM optimization algorithm demonstrates accurate classification performance. When it is used to test the model, 84.99% of classification evaluation accuracy is concluded. Besides, the 0.8924 of AUC that comes from the ROC curve shows a better performance of this model.

Chapter 4. LC Decision Making and Trajectory Planning

4.1 Introduction

This chapter describes the critical problems of LC decision-making and trajectory planning by applying the Markov Decision Process (MDP). Definitions of discrete state and action spaces of LC agent vehicles are made here. Based on that, reward functions are also defined in detail. To coordinate with the simulation via Model Predictive Control (MPC) framework, some redefinitions based on kinematic models are used to transmit vehicle trajectories from the previous machine learning output. Specifically, some considerations with smoothing discrete decisions in the time series are also illustrated in this chapter.

4.2 Problem Statement

4.2.1 Decision Making of AV's LC

Safety is the most important consideration when design a AVs' LC control system. Considering that, the AVs' state at each time space is defined as three levels: safety, attentive, and dangerous. The target risk state of AVs' illegal LC maneuver, human-like central driving habitats, and road boundaries constrain are calculated. Through comparing the risk reward function and the risk assessment value at each time interval, the expected risk of each agent could be defined as the difference between maximum risk assessment value and the reward risk at the next time point.

Since the Risk Assessment (RA) is focused, AV's safety state could be scored as $\{safety = 2, attentive = 1, dangerous = 0\}$ based on expert knowledges. That means the reward from environment could be 2 when it is identified as a safety action that plays an excellent safety control for vehicles driving. Attentive scores as 1, because it is less expected than the safety state, but much better than the dangerous state.

4.2.2 Trajectory Planning of AV's LC

Although the previous deep RL based (most are model-free) models gain a great improvement on LC safety control, they are hard to apply for other scenarios. Considering that, a strong robust LC safety control model of AV should be taken more efforts. Helped by the advanced research on RL, the model-based studies on trajectory planning are becoming more and more popular. As one of typical models, Inverse RL based on Behavior Cloning (BC) is used in this project to infer humans driving behaviors.

From the perspective of personal stylized LC automated safety control, the main problems of these trajectory planning problems are to find an optimized reward function that could best explain the human driving behavior from NGSIM I-80. The other most critical issue is to find the optimal path and control trajectory which could maximize the reward for the platooning vehicles.

4.3 Reward Function for Decision Making

4.3.1 Primary Definition

The risk assessment function is the sum of three items illegal punishment, human driving habits revision, and road boundary limitation. Specifically, the illegal punishment is considered for the exit traffic rules. This is a soft penalty for the AVs that make an illegal lane change. Nearly all human drivers would like to drive in the center of the lane. Hence, the second item is about human driving habits that ensure all AVs driving in the center of the lane. And the last item is driving in the road boundaries and surviving in the environment if possible.

$$r \stackrel{\text{def}}{=} r_{invasrion} + r_{center} + r_{exist} \tag{1}$$

$$r_{invasion} \stackrel{\text{def}}{=} -e^{-\frac{(la_{ld}-la_{hv})^2}{2\sigma^2}} \tag{2}$$

$$r_{center} \stackrel{\text{def}}{=} e^{-\frac{(la_{center}-la_{hv})^2}{2\sigma^2}} \tag{3}$$

$$r_{exist} \stackrel{\text{def}}{=} \begin{cases} 0.1, if\ exist \\ -1, otherwise \end{cases} \tag{4}$$

Where, la_{ld} and la_{hv} denote the lane boundary and the lateral position of the HV, respectively. la_{center} represents the current lane center. Exist means that collision and running out of boundaries do not occur. The values of 0.1 and -1 are commonly used.

4.3.2 Redefinition

Since the risk level defined in this study is a discrete set, it should be transformed into a continuous risk function. As referred from the reproduce project, a risk coefficient τ is induced by the posterior probability that is calculated by Bayes theory. The discrete transfer to continuous risk by the risk coefficient could be written as:

$$\varepsilon \stackrel{\text{def}}{=} \mathbb{E}(\tau) = \sum_{\tau\in\Omega} \tau \cdot p(\tau|d) = \sum_{\tau\in\{2,1\}} \tau \cdot p(\tau|d) \tag{5}$$

Where τ is the discrete risk level {2,1,0}, and ε denotes the expectation of the assessed risk.

4.4 Reward Function for Trajectory Planning

The deep inverse RL based on the Costmap inference is used in this research to help automated safety LC control via MPC. In the beginning, the Costmap data that comes from the NGSIM I-80 data is extracted at each timestamp for deep learning. While the Maximum Entropy Deep Inverse Reinforcement Learning (MEDIRL) is developed and used to find optimal solutions during each control time interval and next generations' updating.

The motivation of learning a spatiotemporal Costmap is that the Costmap obtained from the original MEDIRL cannot be used by itself in MPC. Without temporal information, there are an infinite number of ways to follow the low-cost region in the position Costmap, many of which may cause a collision.

The definition of Costmap is a 2-D bird eye view, which is the occupancy grid of each feature. To describe the LC trajectory smoothly, the vehicle occupancy, velocity occupancy, acceleration

occupancy, heading angle occupancy, offset from the closet lane occupancy are used, which could be shown in Figure 4.1.

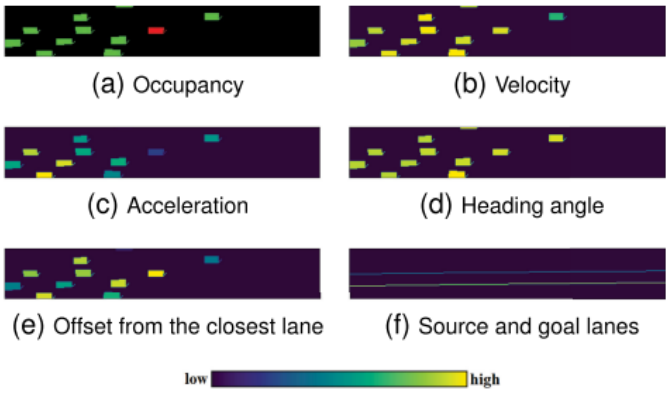

Figure 4.1 The observations that are normalized and converted into the bird's eye view ego-centric 2d images. the brighter color represents higher values as shown in the color bar.

4.5 Summary

The decision-making and trajectory planning of AVs' LC is assumed to follow the MDP in this chapter. Because of the complicated dynamic procedure of LC, the problem statement of the RL-based models is abstracted reasonably in this chapter under MDP assumptions. It is noticed that some redefinitions based on kinematic models are used to transmit vehicle trajectories from the previous machine learning output because of the MPC control framework that is used to simulate this process.

Chapter 5. Optimization Methodology

5.1 Introduction

This chapter will discuss how to find the optimal policy to describe the ground truth under the assumption of behavior cloning. The mechanism of two typical reinforcement learning models, Prioritized Replay Deep Q-Network (PRDQN) and MEDIRL, are illustrated. Considering that, the general optimization framework is formulated. The objective function will be defined to minimize the risk level or collision of LC agents and best mimic the human driving data as described in Chapter 3.

5.2 Definition of Reinforcement Learning

RL is used to maximize the numerical reward for the action of the decision agent by learning from strategies produced by the interaction between the agent and environment. As for the most interesting and challenging cases, the agents' actions not only reflect the immediate reward but also the subsequent reward for the following steps. The mechanism of RL is described as a Markov Decision Process (MDP) in the most common situation. The PRDQN is based on the basic principle of reinforcement learning MDP process, which can be written as:

$$\mathcal{M} \Rightarrow \perp def < \mathcal{L}, Å, \wp, \mathcal{R} > \quad (1)$$

Where $\mathscr{L}\mathcal{L}$ denotes a finite set of states, Å is a finite set of actions, $\wp$ represents the state transition probability, and $\mathscr{R}\mathcal{R}$ denotes the reward space.

Typically, as for a specific environment state, a stochastic policy is generated which is the probability of action that is taken. During these interactions, the set of these state-action mapping samples could be obtained. Among them, a best policy function exists and can be written as follows to maximize the expected cumulative reward:

$$\pi^*(s) = {}^{argmax}_{\pi} E_\pi\left\{\sum_{i=0}^{+\infty} \Upsilon^i r_{t+i} \,|s_i = s\right\} \quad (2)$$

Where $\Upsilon \in [0,1]$ denotes the discount factor, which controls the weight of the future reward, $\pi^*(s)$ represents the best policy, and r_{t+i} is the reward of time $t+i$, which can be calculated by the pre-defined reward function.

5.3 Definition of PRDQN

The RL optimization control solved by the PRDQN algorithm which is based on MDP has been proved to be converged when its reward function is a bounded one. Figure 5.1 shows the basic architecture of this study.

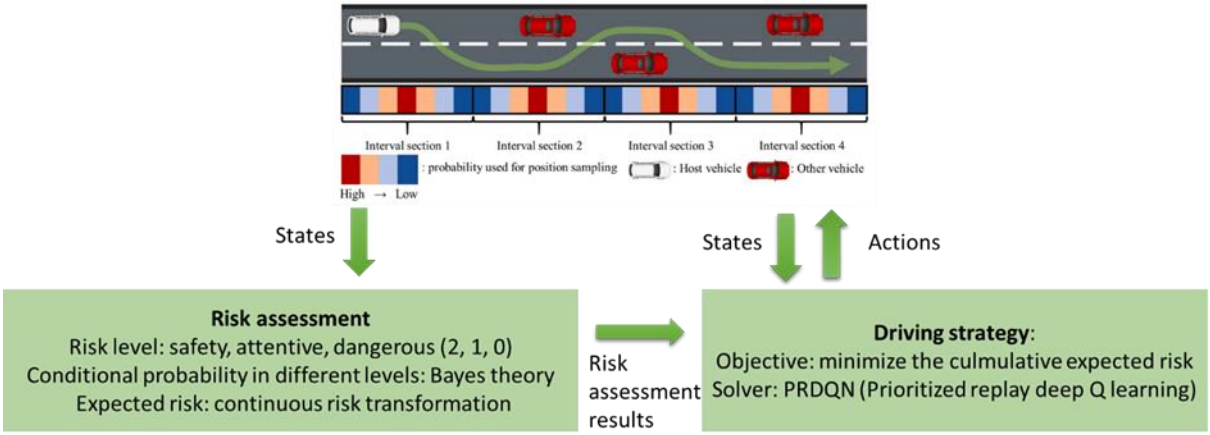

Figure 5.1 Structure of RA-PRDQN

5.3.1 **Principle of PRDQN**

As for DQN, the Q value function is induced to solve the above-mentioned question. As for each best policy function, the corresponding Q-value function could be written as:

$$q^{\pi}(s,a) = E_{\pi}\{\sum_{i=0}^{+\infty}\gamma^{i} r_{t+i} | s_i = s, a_t = a\} \tag{3}$$

Where $q^{\pi}(s,a)$ denotes the expected cumulative reward starting from state s following policy π and action a.

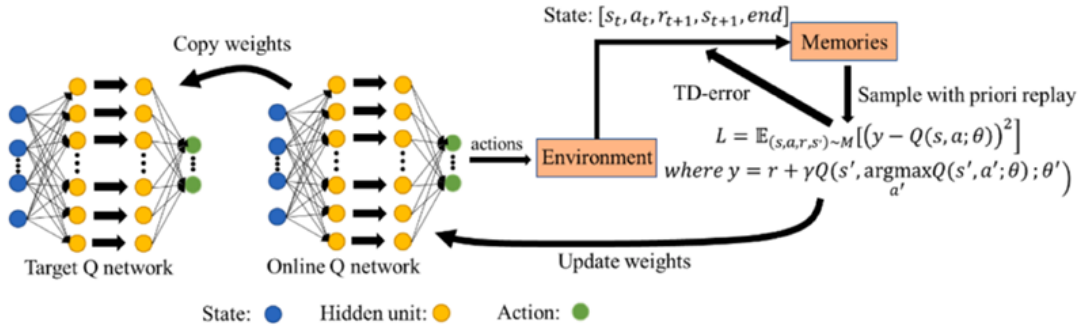

Figure 5.2 Typical architecture of PRDQN

The structure of a typical PRDQN model is shown in Figure 5.2. From this, the most significant difference of PRDQN and other DQN models is the operation for TD error. The samples with small TD errors are easy to be learned, but it is not the case for the samples with higher TD errors. The prioritized replay is inducted to improve the probability of higher TD errors for learning process, which is defined as:

$$p(i) \overset{\text{def}}{=} \frac{p_i^a}{\sum_k p_k^a} \tag{4}$$

Where $p(i)$ is the sampled probability of sample i, p denotes the TD error, and a is a pre-defined coefficient.

In order to guarantee the priority of higher TD error samples, the weighted samples are used to adjust the gradients when updating the network for bias elimination. It could be written as:

$$w_i \overset{\text{def}}{=} (\frac{1}{N} \cdot \frac{1}{P(i)})^{\beta} \tag{5}$$

Where N is the replay size, and β is a pre-defined coefficient.

As for the expected risk function has transferred from the discrete one to the continuous one, the corresponding Q-value function could be written as:

$$Q^{\pi}(s,a) = \quad E_{\pi}\{\sum_{i=0}^{+\infty} \Upsilon^{i}(max\varepsilon - \ \varepsilon_{t+i}) \,|s_i = s,\ a_t = a\} \tag{6}$$

Where $max\varepsilon = 2$.

Therefore, the purpose of PRDQN is to search for an optimized strategy by Q function to minimize the overall expected risks of AVs.

Compared with Q-learning, there are two main improvements of DQN. First, the value function is estimated by the deep Convolutional Neural Network (CNN). Second, the learning process is trained by experience replay as well.

5.3.2 Training Details of PRDQN

The most important part of training process is to ensure the PRDQN algorithm to find an optimal policy with the minimum expected risk. This study still uses five layers with variable hidden units in each to train the deep learning structure, and the activation function is determined by ReLu and Linear for the output. The details of the network are shown in Table 5.1.

Table 5-1 Details of the Q-value network

Layers	Hidden Units	Activation Function
1	30	ReLu
2	112	ReLu
3	304	ReLu
4	120	ReLu
5	34	ReLu
Output	5	Linear

Specifically, there are a lot of training tips for achieving a significant training result. The fully connected layer is utilized to approximate the Q-value function. In addition, Batch Normalization (BN) is utilized to normalize the neural output before activation.

Warmup learning rate strategy is used to reduce the variance when updating the network, in which the initial learning rate is 0.01 and recovers to 0.1 after 50 episodes.

Gradient clipping is used to clip the gradient by normalization. The purpose of this is to avoid gradient explosion for a complicated huge network. For any gradient in layer, the gradient after clipping could be defined as:

$$grad_i^* \overset{\text{def}}{=} grad_i * \frac{clipnorm}{max(norm(grad_i), clipnorm)} \tag{6}$$

Where $grad_i$ and $grad_i^*$ denote the raw gradient and the gradient after clipping in layer I respectively; $norm$ denotes the standard deviation calculation; and $clipnorm$ is a pre-defined coefficient which denotes the standard deviation after clipping. To reduce the updating variance, $clipnorm$ is set as 0.1.

Soft network updating: Rather than using the hard network updating, soft updating is used to copy the weights from the online network to the target network, which can be written as:

$$\theta_{target} \stackrel{\text{def}}{=} (1-\eta)\cdot\theta_{target} + \eta\cdot\theta_{online} \tag{7}$$

Where θ_{target} and θ_{online} denote the weights in the target network and the online network respectively, and η is a small value that affects the speed of the target network updating and it is set as 0.

5.4 Inverse RL

5.4.1 Maximum Entropy Inverse RL

Given the expert's demonstrations, if the expert's behavior is suboptimal (imperfect or noisy), it is hard to represent the behavior with a single reward function. The Maximum Entropy Inverse RL approach is introduced to solve this ambiguity problem.

Maximizing the entropy of distributions over paths while satisfying the feature expectation matching constraints is equivalent to maximizing the likelihood of the observed data D under the assumed maximum entropy distribution:

$$\theta^* = \underset{\theta}{argmax} \sum_{\varsigma\epsilon D} logP(\varsigma|\theta, P_{sa}) \tag{8}$$

Where $P(\varsigma|\theta, P_{sa})$ follows the maximum entropy (Boltzmann) distribution. This convex problem is solved by gradient-based optimization methods with

$$\frac{\partial L(\theta)}{\partial\theta} = \sum_{s\epsilon\varsigma\epsilon D}\mu_s f(s) - \sum_{s_i}\mu_{s_i} f(s_i) \tag{9}$$

Where μ_s is defined as the State Visitation Frequency (SVF), the discounted sum of probabilities of visiting a state s: $\mu_s = \sum_{t=0}^{\infty}\gamma^t P(s_t = s|\pi, \theta, P_{sa})$. With a given or selected f, this updated rule ends up as finding θ in reward in which an optimal policy matches the SVF of the demonstration D.

5.4.2 MEDIRL

Previous approaches to estimate a reward function used a weighted linear reward function with hand-selected features. To overcome the limits of linear expression, a new method towards to Neural Network regression has been proposed to extend it to the nonlinear problem $R_\theta(s) = R(f(s), \theta)$. By training a NN with a raw observation obtained from sensors as an input, it does not require hand-designing state features.

In MEDIRL, the network is trained to maximize the joint probability of the demonstration data D and model parameters θ under the estimated reward $R_\theta(s)$:

$$L(\theta) = logP\big(D, \theta\big|R_\theta(s)\big) = logP\big(D\big|R_\theta(s)\big) + logP(\theta) = L_D + L_\theta \tag{10}$$

Since L_θ can be optimized with weight regularization techniques for training Neural Networks, MEDIRL focuses on maximizing the first term L_D:

$$\frac{\partial L(\theta)}{\partial \theta} = \frac{\partial L_D}{\partial R_\theta}\frac{\partial R_\theta}{\partial \theta} = (\mu_D - \mathrm{E}[\mu])\frac{\partial R_\theta(s)}{\partial \theta} \tag{11}$$

Where $\mathrm{E}[\mu]$ is the expected SVF from the predicted reward. In the MEDIRL update, the derivative of the reward with respect to the weight parameter can be easily computed by back-propagation.

5.5 Summary

This chapter presents the optimization used in this research. Two critical problems are solved based on the RL algorithm. One is to find the optimal reward function by the Q value of PRDQN to minimize the driving risk during LC. The other is to find the optimal policy which could demonstrate humans' driving characteristics and the optimal value of MEDIRL that could ensure the maximum safety reward during LC. It is noticed that only locally optimal solutions can be obtained due to constrained generations. With more generations, the solution can be further improved to approach closer to the global optimal.

Chapter 6. Simulation and Validation

6.1 Introduction

This chapter describes the basic settings of the automated LCs control process in CARLA. As one of the most popular open-source traffic simulation software, CARLA provides various digital maps, vehicle simulation controllers, and multiple APIs that could be conveniently connected to other toolboxes. Additionally, some typical simulation scenarios of dense traffic set in CARLA are illustrated in this chapter.

6.2 Simulation Platform

6.2.1 CARLA

The development of artificial intelligence, 5G, and V2X produces a high possibility of AV application in the market. The implementation of AV should be based on a huge amount of training data. The validation of different driving scenarios plays an important role as well. However, it is unrealistic to test with real vehicles for a large amount of training data and environmental verification, because it involves huge safety costs and implementation cycle length. Most important, many scenes cannot be reproduced in the real environment, such as the vehicle in front suddenly losing control. CARLA can meet researchers' high expectation on a mimic simulation for AVs' control and evaluation.

CARLA is an open-source project jointly developed by Intel LABS and the Computer Vision Center in Barcelona. The framework of Carla is a server/client architecture. The server is the simulation environment itself. All the things that are present in the real driving environment of a car are embodied in it, including: cars, people, roads and Bridges, traffic lights, signs, weather, buildings, etc. Based on these components, CARLA could simulate a real world for AVs' algorithms training based on Python libraries.

The development and update of CARLA rely on a strong research team behind it. Hence, CARLA provides multi-versions to suit different machine learning frameworks on various operating systems. The front end of CARLA is presented by Unreal Engine for 3D rendering. Two versions (Linux and Windows) are available right now for clients based on powerful GPU.

The purpose of this research is to test the performance of the proposed LC control model and trajectory planning model of AVs' platoon. The Linux version is applied because of its higher reliability on calculation than the Windows version. In general, the proposed models are coded by Python that interact with CARLA by API. The LC decision is based on the reward learning from the interaction of AVs' actions and the safety of the platoon. The trajectory is periodically transmitted to the CARLA Server. Every time the play button is clicked; the simulation could be conducted.

However, the definition of Python API is not easy to make. Fortunately, the Server side has *A Wheeled Vehicle AI Controller* and autopilot model. The mechanism of the *Set Fixed Route ()* function shows a perfect example of how to control AVs' step-by-step operations by some

inner customized modules. Based on that, the revised Python API for the proposed models in this research can be coded. Finally, it needs to connect the defined API with the server and client.

6.2.2 CARLA Setting

Initially, it is assumed that the LC maneuvers of automated vehicles are controlled by following the MDP. One significant benefit of this assumption is that the PRDQN could be utilized to explore the nearly optimized decision at each time step as soon as possible. However, some calculation problems would occur if there are some unreasonable definitions of this system. The kinematic information is usually utilized like the end-to-end control problem. Since PRDQN model was used in this study to train the data, the numerical information could be easily obtained. And the parameters could be easily set in CARLA as well. The numerical information is used below:

$$VAO_i = [\mathbb{i}_i, lO_i, la_i, yaw, \Delta lO_i, \Delta la_i, \Delta yaw] \quad (1)$$

$$s = \left[VAO_1, VAO_{2,} \ldots VAO_n\right] \quad (2)$$

$$\mathbb{i}_i \, \epsilon \, [0,1], 0 \leq i \leq n \quad (3)$$

Where $\mathbb{i}_i$ denotes whether there is another vehicle in lane i within the perception range, n is the number of lanes, ΔlO_i and Δla_i are the corresponding change rate, and yaw and Δyaw denote the vehicle yaw angle and yaw rate.

The corresponding actions of each AV could be described by the following set:

$$a_t \stackrel{\text{def}}{=} [LTL_t, LTS_t, S_t, RTS_t, RTL_t] \quad (4)$$

Where LTL and RLT denote left-turn and right-turn with a large numerical value respectively, LTS and RTS denote left-turn and right-turn with a small numerical value respectively, and S denotes straight driving action without steering.

6.2.3 Parameters Setting

In this project, the proposed LC decision-making and trajectory planning model are tested in a mixed traffic flow, which is combined by the Human-driven Vehicles (HV) and AVs. The lateral movements of AVs are controlled by RL model. The longitudinal movements of AVs are controlled by Intelligent Driver Model (IDM). The Human-driven Vehicle (HV) dynamics used in this research are shown in Table 6-1.

Table 6-1 Brief information on the HV dynamics

Parameter	Description	Value
Max rpm	The maximum RPM of the vehicle engine	5000.0
Moi	The moment of inertia of the vehicle's engine	1.0
Clutch strength	The clutch strength of the vehicle in Kgm2/s	10.0
Final ratio	The fixed ratio from transmission to wheels	4.0
Mass	The mass of the vehicle in Kg	1000.0
Drag coefficient	Drag coefficient of the vehicle's chassis	0.3
Steering curve	Curve that indicates the maximum steering for a specific	-

	forward speed	
Tire friction	A scalar value that indicates the friction of the wheel	2.0
Damping rate	Damping rate of the wheel	0.25
Max brake torque	Maximum brake torque in Nm	1500.0
Max handbrake torque	Maximum handbrake torque in Nm	3000.0

6.3 Scenarios

6.3.1 Dense Traffic Scenario

LC in dense traffic has been a challenging application in autonomous driving, especially in scenarios with complex inter-vehicle interactions. It is a common safety-efficiency dilemma. Some planners have larger buffer space to handle uncertainties from surrounding vehicles and the environment, thus can be overly conservative and inefficient. While other planners put more emphasis on efficiency and task success rate, safety is compromised. It is more challenging during the transition period to a fully automated transportation system because HVs and AVs need to interact with each other and share the transportation network. So, without an accurate estimate of the other vehicle's intention, the LC process can be neither inefficient nor unsafe.

Due to the high LC frequency, the simulation scenarios are challenging. For one reason, the agent vehicles would take dense actions for the static objectives and the relatively high speed for the moving objectives. For the other reason, it is essential to define the action space and state space of each agent vehicle in this research based on RL. The computation efficiency and time efficiency would be very low if there are too many action and state values that are used to explore the optimal reward function. Since the MEDIRL mentioned in chapter 4 would be applied to plan LC agents' trajectory, it would explore more than one optimal reward function at each time step. When the output of MEDIRL is used for MPC control as the input, there would have more than one actions to control the agent. It is easy to make the agent vehicle fall into a local optimum.

6.3.2 Simulation of Decision Making

In order to evaluate the effectiveness of the proposed RA-PRDQN model in this study, the scenarios that consider the fixed obstacles and moving obstacles should be considered comprehensively. Constrained by the strong assumptions of this model, this study intends to lay the longest straight road section by the CARLA default town map. Thus, the randomly static and dynamic vehicles are placed on a 420m long straight road. As shown in Figure 6.1, a typical description of LC phenomenon is based on this proposed model.

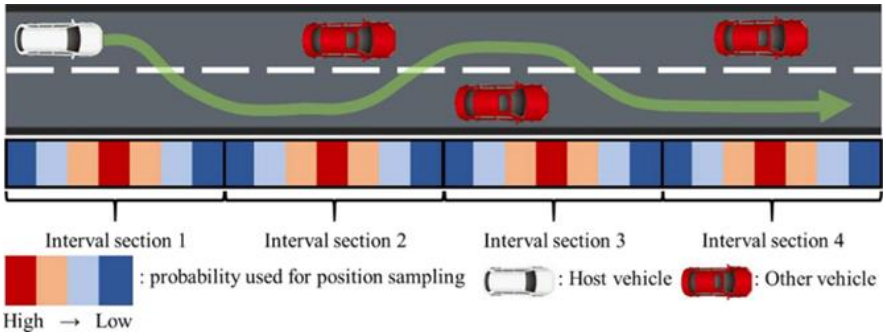

Figure 6.1 Layout of LC trajectories on CARLA

6.3.2.1. Static Vehicles

As shown in Figure 6.1, the red vehicles are randomly placed on the above-mentioned straight road and the quantity of these static vehicles ranges from 10 to 26. Since the 420m straight main freeway is relatively short, 100 times during the simulation process are executed to gain a better simulation result. The purpose of that is also used to ensure about 4 times per 100m (high lane change frequency for on-road driving). The location of the static vehicle in each interval section is randomly specified using the Gaussian-based position sampling method, and the LC of the initial locations is randomly set. Therefore, the situation with two vehicles parallelly located in two lanes to block the road would not happen in our experiments.

6.3.2.2. Moving Vehicles

Since the vehicles in the scenario1 are fixed, it is easy to set these static ones. However, the moving obstacle of human driving vehicles is slightly different from the fixed one. In this study, the human-driving vehicles are dynamically set as shown in Table 4-1. And the HVs are set in the autopilot mode provided by CARLA (speed limit 30 m/s).

6.3.3 Simulation of Trajectory Planning

In this section, the experiments with the CARLA simulator with ROS is applied to test the trajectory planning model based on the MEDIRL. The scenario in this section is similar to the simulated decision-making. A dense traffic highway scenario with 20 vehicles driving around the ego vehicles is designed as well. Other vehicles perform lane keeping and collision avoidance and each vehicle tries to reach their target speed, which is randomly generated by $5 + U[-2,2]$ in m/s, where U is uniform sampling.

The behavior model of the other vehicles follows the IDM, one of the well-known rule-based models for car following. The model is also based on the bicycle kinematics. The principle of this kinematic model is:

$$x_{k+1} = x_k + v_k \cos(\psi_k + \beta_k)\Delta t \quad (1)$$

$$\psi_{k+1} = \psi_k + \frac{v_k}{l_r}\sin(\beta_k)\Delta t \quad (2)$$

$$y_{k+1} = y_k + v_k \sin(\psi_k + \beta_k)\Delta t \quad (3)$$

$$v_{k+1} = v_k + v_k \cos \Delta t \quad (4)$$

$$\beta_k = \tan^{-1}(\frac{l_r}{l_f+l_r}\tan(\delta_k)) \quad (5)$$

Where a and δ are the control inputs: acceleration and the front wheel steering angle. β is the angle of the current velocity of the center of mass with respect to the longitudinal axis of the vehicle, (x, y) are the position, the coordinates of the center of mass in an inertial frame (X, Y) , ψ is the inertial heading angle, and v is the vehicle speed. l_r and l_f are the distance from the center of mass to the front and rear of the vehicle, respectively. The state is defined as $[x, y, \psi, v, \beta]$.

The other vehicle's behavior is designed to be always cooperative, where they slow down if the AVs crosses a line in front of them and cuts into their lane. It is performed 50 experiments per algorithm at each trial, and the environment is randomized by starting with a different initial velocity of the AV and relative initial positions and target velocities of other vehicles.

To reduce the gap between the real vehicle's model in CARLA and the kinematic bicycle model adopted in MPC, this study publishes the MPC-predicted state trajectories as waypoints $[x, y, v]$. Then a low-level PID controller executes the vehicle's control commands (throttle and steering angle) to follow the MPC-generated waypoints.

6.4 Summary

The new technology CARLA applied in this research to simulate decision-making and trajectory planning models that have been illustrated in the aforementioned chapters. CARLA is a friendly used simulation tool for CAVs because of its opening modification modules. Helped by the previous studies of vehicle dynamics' validation, the basic setting of vehicles, platooning, and simulation environments are illustrated in this chapter. The classic IDM model is applied to control longitudinal maneuver and the lateral movement controlled by the proposed model. Besides, the settings of dense traffic scenarios are demonstrated here for further robust analyses.

Chapter 7. Numerical Results and Discussion

7.1 Introduction

This chapter will present a summary of the simulation results. The results of the four simulation scenarios are discussed in detail.

Since the AV's LC model is abstracted into MDP, the optimization solver PRDQN and Inverse RL are applied to solve decision making and trajectory planning as mentioned in Chapters 4 and 5. Hence, the reward function will be estimated, considering maneuver safety and control efficiency. For that, the collision number, statistical analysis of gap distance, acceleration, and jerk angle of each scenario are estimated with different combinations of HVs and AVs, and the effects of the LC success rate of AVs could be quantified.

7.2 Result of SVM

7.2.1 Quantitative Results

Finally, 473 groups of test samples are imported into the SVM model for intention recognition, according to the screening results. The classification results are shown in Figure 7.1, where the x-axis is the total number of test samples, and the y-axis is the classification result (0-lane keeping, 1-lane changing). Of the test samples, 402 groups have completed the accurate identification of driving behavior, and the classification accuracy is 84.99%.

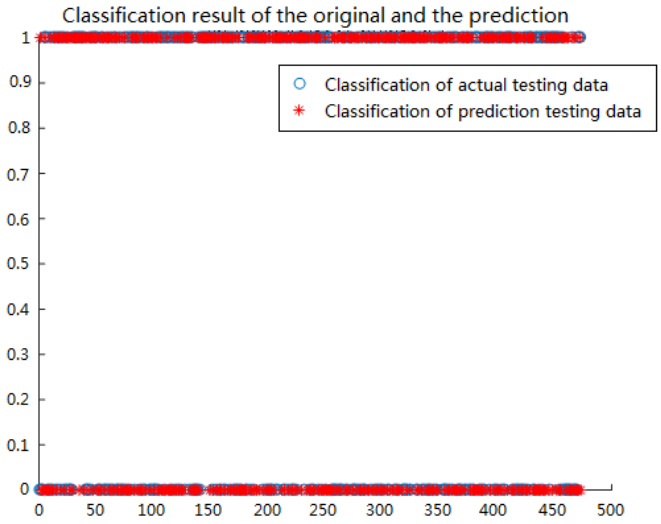

Figure 7.1 Identification results of SVM

By comparing the actual intention of the driver in the test set with the recognition result of the model, the recognition accuracy of LC and lane-keeping are 87.61% and 84.17%, respectively, as shown in Table 7-1.

Table 7-1 Lane changing and lane keeping identification accuracy

Actual intention	Identification intention		Identification accuracy
	Lane changing	Lane keeping	
Lane changing	99	14	87.61
Lane keeping	57	303	84.17

7.2.2 Discussion of Intention Identification

In addition to the classification accuracy of the analysis model, the ROC will be used in this chapter to further verify the model performance. ROC has high reliability and accurate description, is not affected by noise data, and has been widely used in the field of machine learning. ROC reflects the relationship between *Sensitivity* and *Specificity*, and a curve is plotted using *1-Specificity* as the abscissa and *Sensitivity* as the ordinate. The theory of ROC is shown as:

$$\text{sensitivity} = \frac{TP}{TP+FN} \times 100\% \quad (1)$$

$$\text{specificity} = \frac{TN}{TN+FP} \times 100\% \quad (2)$$

Where, TP (the true positive category) is the number of LC vehicles that is properly identified.

FN (the fake negative category) is the number of no LC vehicles that are identified as LC maneuver.

TN (the true negative category) is the number of LC vehicles that is identified as no LC maneuver.

FP (the fake positive category), is the number of no LC vehicles that is identified as LC maneuver.

Therefore, the models demonstrate more accurate separating performance when the value of *sensitivity* is significantly greater than *1-specify*.

The ROC result of LC intention identification after testing is shown in Figure 7.2. It is obvious that the position of ROC curve is closer to the top left corner. That demonstrates that the *sensitivity* of the proposed SVM model is much greater than *1-specify*, indicating an accurate classification.

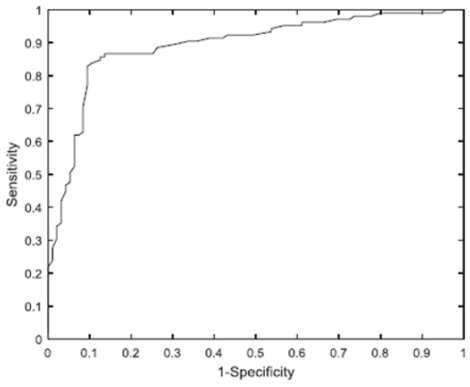

Figure 7.2 The ROC curve of model testing

To evaluate the models' performance quantitatively, the ROC Area under Curve (AUC) is used to calculate the decision value of identification algorithm after self-learning. Usually, AUC is calculated by the probability of the positive one in front of the negative one when randomly selecting a positive sample and a negative sample from the test set.

Figure 7.3 describes the relation between ROC and AUC. From that, it is clear that the value of AUC is always smaller than 1. The closer its value is to 1, the more accurate recognition and classification can be carried out. So, the index is adapted to evaluate the performance of SVM for lane changing intention identification. The results are shown in Table 7-2.

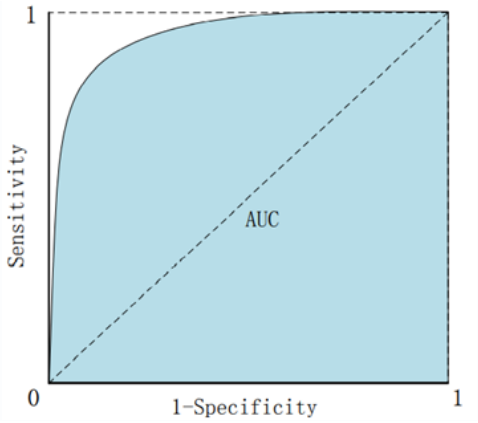

Figure 7.3 Mechanism of ROC and AUC

Table 7-2 AUC model performance evaluation standard

AUC	Model performance evaluation
0.9~1.0	A
0.8~0.9	B
0.7~0.8	C
0.6~0.7	D
0.5~0.6	E

According to the testing result of ROC curve that is shown in Figure 7.2 and the corresponding AUC based on Table 7-2, the testing result is 0.8924 and the final rating result is B.

Regarding the number of fake positive samples and fake negative samples, which influence the model accuracy most, the fake positive samples that misidentify the lane keeping to lane changing dominate the ratio. Drivers usually will cancel their land changing intention when some unexpected incidents occur suddenly. Therefore, an independent T test is used in this chapter to discuss the dominant feature to identify the two samples: misidentification of lane keeping to lane changing and misidentification of lane changing to lane keeping.

Table 7-3 T-test results of independent samples' features

Features	Classification	Mean	Significance test ($\alpha = 0.05$)	
v_x(m/s)	0-car following	10.71	$p = 0.073 > 0.05$	Non-significant
	1-lane changing	10.48		
v_y(m/s)	0-car following	0.07	$p = 0.000 < 0.05$	Significant
	1-lane changing	1.31		
a_x(m/s^2)	0-car following	0.18	$p = 0.259 > 0.05$	Non-significant
	1-lane changing	0.15		
a_y(m/s^2)	0-car following	0.33	$p = 0.344 > 0.05$	Non-significant
	1-lane changing	0.38		
θ(deg)	0-car following	0.10	$p = 0.002 < 0.05$	Significant
	1-lane changing	5.67		
Δv_1(m/s)	0-car following	0.28	$p = 0.749 > 0.05$	Non-significant
	1-lane changing	0.19		
Δv_2(m/s)	0-car following	0.24	$p = 0.494 > 0.05$	Non-significant
	1-lane changing	0.29		
Δv_3(m/s)	0-car following	0.27	$p = 0.177 > 0.05$	Non-significant
	1-lane changing	0.35		
Δd_1(m)	0-car following	8.74	$p = 0.202 > 0.05$	Non-significant
	1-lane changing	10.13		
Δd_2(m)	0-car following	10.97	$p = 0.246 > 0.05$	Non-significant
	1-lane changing	7.70		
Δd_3(m)	0-car following	11.18	$p = 0.328 > 0.05$	Non-significant
	1-lane changing	9.70		

From the T testing results shown in Table 7-3, the dominant two features are the current vehicles' lateral speed (v_y) and the turning angle of steering wheel θ(deg). Comparing the average lateral speeds between them, the state of car-following is significantly larger than the lane changing. The data indicates that it is unreasonable to identify the lane changing intention merely through vehicles' lateral speed at the current moment. Since the lane changing is a smooth and continuous lateral moving process, it is more scientific to analyze the lateral speed variance in a certain period not just an instantaneous one.

As for the turning angle of steering wheel, it is 0.01 deg for both the car-following and lane changing when it is straight driving. However, it turns to 5.67 deg when starting to LC, which means that the driver takes a lateral adjustment at some degrees.

Therefore, to identify the lane changing intentions more accurately, a second judgment about the lateral average speed and turning angle of steering wheel needs to be done and analyzed. The project recommends more research and discussions related to the precise lane changing identification to be taken when more precise data are supported, which plays a significant role on safety driving of automobile study in the future.

7.3 Numerical Results of Decision Making

7.3.1 Quantitative Results

After training the PRDQN model and RA-PRDQN model on CARLA, the quantitative results when using different methods in those two scenarios are shown in the following Table 7-4, Figure 7.4, Table 7-5, and Figure 7.5. The baseline means a random action strategy, denoting the difficulty level of the experimental scenario. Score ($\boldsymbol{\mu}$) and Score ($\boldsymbol{\sigma}$) denote the average score (travel distance) and the corresponding standard deviation in the examined scenarios, respectively. nC_s denotes the number of collisions with vehicles or road boundaries occurring in the experiments. Δ denotes the relative change rate of proposed model than the random model.

Table 7-4 The metrics when using different methods in scenario 1

Method	Score ($\boldsymbol{\mu}$)	Δ（%）	Score ($\boldsymbol{\sigma}$)	Δ（%）	nC_s	Δ（%）
Baseline	30.7	-	10.4	-	98	-
PRDQN	321.6	1047.6 up	139.2	133.8 up	55	56.1 down
RA-PRDQN	399.7	124.3 up	35.6	25.6 down	8	14.5 down

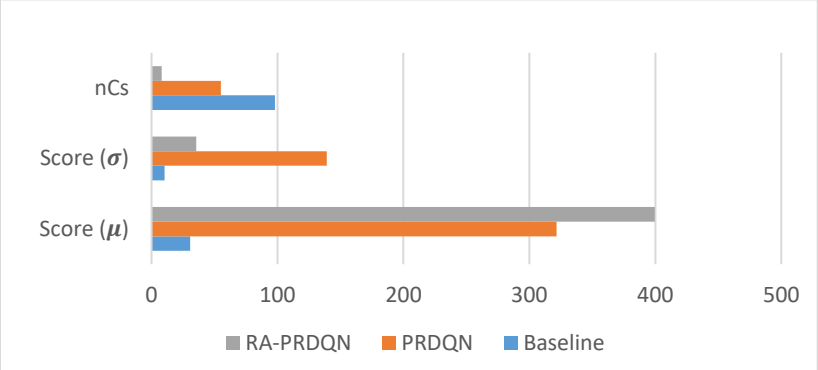

Figure 7. 4 Comparison Result Among Models in Scenario 1

Table 7-5 The metrics when using different methods in scenario 2

Method	Score ($\boldsymbol{\mu}$)	Δ（%）	Score ($\boldsymbol{\sigma}$)	Δ（%）	nC_s	Δ（%）
Baseline	33.1	-	15.6	-	100	-
PRDQN	258.6	781.3 up	100.6	644.9 up	57	57.0 down
RA-PRDQN	403.2	155.9 up	72.1	57.2 down	10	17.5 down

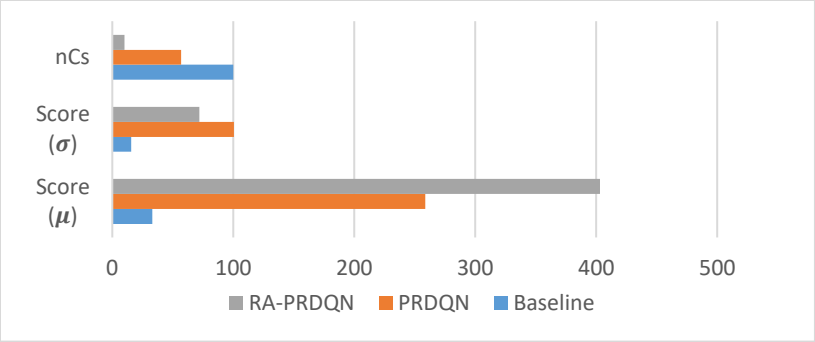

Figure 7.5 Comparison result among models in scenario 2

The presented results show that the proposed methods can attain better scores than the original methods. Specifically, the average scores of PRDQN in scenario 1 and scenario 2 are 321.6, 258.6, which are improved by 1047.6%, 781.3% compared with the randomly selected baseline. This indicates that the DRL based model could greatly improve LC success rate to make sure a stable travel distance of AVs. The obvious decrease in the number of collisions also proves this conclusion.

When comparing the PRDQN and proposed RA-PRDQN model, the average score shows similar raising tendency. The score of RA-PRDQN is 403.2, which is improved by 155.9% over PRDQN. As for the collision, only 10 occurred during the simulation test, which is much less than the PRDQN model. This indicates that the LC models achieve more stable performance after introducing the RA reward function into the PRDQN model.

In summary, the presented quantitative results show that the proposed RA-PRDQN model can achieve a superior performance in both the static obstacle scenario and the scenario with dynamically moving vehicles. It indicates that the risk awareness strategy could help the agent be aware of the dangerous actions that may lead the vehicle into collisions during simulation. Introducing the risk awareness in PRDQN will make the agent to be punished when any risk behavior is taken. Hence, the proposed RA reward function will be better in reflecting the potential risk in complex driving scenarios, and more adaptive to take reasonable actions to avoid near-collisions and reach a more stable performance.

7.3.2 Result Discussion

The simulation scenarios include a static vehicle scenario and a moving vehicle scenario. In fact, the static test scenario of automated LC model is much easier than the dynamic one, because of much more uncertainties of dynamic systems. However, due to the high lane change frequency, the static scenario is still challenging.
Table 7-6 shows the comparisons of the simulated driving parameters between this research work and the previous studies. It could be concluded that the simulated driving task in this study is challenging even for a static one. In the static scenario, the agent needs to make about 16 lane changes in each evaluation episode. In the moving scenario, the agent should make about 6 lane changes with a relatively high speed (roughly between 10 ~ 20 m/s). Therefore, the examined lane change scenarios in the present study are more challenging as compared with the lane change scenarios in the previous studies. In the examined challenging scenarios, the success rate of the proposed RA-PRDQN is about 88 ~ 94% in the examined dense traffic flow scenario with high lane change frequency. Therefore, the success rate of the proposed RA-PRDQN should be within a reasonable and acceptable range.

Table 7-6 The metrics when using different methods in scenario 1

Method	Total experiment distance	Total obstacle number	Mean barrier spacing
Wang et al., 2018	100 m	8	125 m
Mirchevska et al., 2018	1255 m * 10	50 *10	25 m
Ye et al., 2020	800 m	-	-
Duan et al., 2020	800-1000 m	2	450 m
Chen et al., 2020	420 m	2	200 m
This research	420 m * 100	25 * 100	16 m

7.3.3 Discussion of LC Failure

Although the proposed RA-PRDQN has greatly improved the motion path stability and quantitative performance, there are still some failure cases. As shown in Table 7-4 and Table 7-5, there are eight collisions in scenario 1 and ten collisions in scenario 2 when using the proposed RA-PRDQN model. Most of the failure cases are caused by the short longitudinal distance between the front vehicle in the current lane and the lag vehicle in the target lane. One of the causations of these failures is the imperfectness of the used sampling probability method for position initialization of the HV and Objective Vehicles.

Although the used sampling probability method can ensure that the situation with two vehicles parallelly located in two lanes blocking the road would not happen, there would be the situation where two vehicles are on different lanes with very short longitudinal distance (i.e., gap) which is not enough for the HV to execute a lane change maneuver. Therefore, a collision would not be avoidable because the trained agent is targeted to arrive at the destination to complete the driving task. In our future work, the sampling probability method for position initialization will be improved to avoid this problem, based on which the performance of our proposed methods should be enhanced.

Another important reason leading to these failures is that the actions considered in this study for collision avoidance only cover the steering behavior. However, braking behavior is also very important to control the distance to the front and lag vehicles for collision avoidance. The missing of speed adaptation by braking may greatly contribute to those failure cases. To implement more realistic behavioral actions for autonomous driving, our future work will enrich the action space by including the braking behavior for longitudinal control together with the steering behaviors for lateral control.

The third possible causation of these failures may be that the driving style attribute in the designed RA-PRDQN is similar with the aggressive driving style performance. Given that driving style affects drivers' lane change decisions, the future efforts should focus on solutions to addressing this problem by considering driving style preferences in the designed decision-making strategies.

7.4 Trajectory Planning Results

7.4.1 Quantitative Results and Discussion

To test the model's performance, multiple baselines are compared here. They are BC (Res NET18), Prediction and Policy learning Under Uncertainty (PPUU), and (Neural Network) NN MPC. The performance of these models is described in Table 7-7.

Table 7-7 The metrics when using different methods in scenario 1

Model	Time	Succ.(%)	Coll.(%)	Timeout (%)	Brake Avg	Thr. Aug	Acc. Max	Brake Jerk Avg	Thr. Jerk Avg	Ang. Acc. Avg	Ang.Acc.Max	Ang.Jerk.Avg	Ang.Jerk.Max
BC(ResNet18)	14.45	44	56	0	-0.34	0.63	1.55	-0.59	0.68	0.24	1.75	0.45	7.78
PPUU	14.77	24	76	0	-2.86	0.69	1.76	-0.39	1.29	0.17	3.24	0.52	27.49
NNMPC	13.20	86	14	0	-0.74	0.54	1.34	-0.78	1.09	1.80	18.29	3.46	76.47
MEDIRL	25.63	88	10	2	-0.51	0.59	1.72	-0.61	0.69	1.52	16.9	2.89	43.02

Note: Comparative Analysis and Ablation Study (With N=50).

First, the research claims that the BC models are able to finish the task with about 80% of success rates with a simple scenario (Pei and Xu, 2006). In which, AVs are controlled by BC model, while other vehicles run in a constant speed maintaining a large constant gap. In this challenging scenario, BC models are not able to finish the lane changing with more than 50% success.

As PPUU is trained with the entire NGSIM dataset that mostly includes driving straight and because of the small clamping value for action (0.01 rd for steering angle), it mostly drives straight until it crashes to the front vehicle. This result shows how fragile the RL-trained policies are when being tested at a new environment.

The next baseline, NNMPC, is able to achieve 86% successes. Compared to the other baseline models, NNMPC does not only rely on the learned or trained models, and it finds a rule-based optimal solution online on top of the NN-predicted behaviors. Although the NNMPC has strict safety constraints in its optimization, it is believed that the prediction model of other vehicles might fail sometimes when other vehicles' velocity changes frequently.

Next, some MEDIRL related trials has been tested. The Model Predictive Path Integral (MPPI) with the non-temporal Costmap is used to learn with the original MEDIRL algorithm for comparison. Since it cannot extract correct waypoints from non-temporal Costmap, this research did not test the MPC with the non-temporal Costmap. As the non-temporal Costmap does not include any notion of optimal velocity, unlike our spatiotemporal Costmap, the MPPI starting with zero initial velocity does not find an optimal solution to perform a lane change with the non-temporal Costmap. However, once it explores the wrong/opposite direction to the goal lane, the Costmap predicted at a new state (edge-case) is not correct, as the input data it takes has a very different distribution compared to the input in the training data. It should be emphasized that this compounding error problem still exists in Deep IRL and is one of the limitations of the reward function (cost function in MPC) learning methods

that only learn from successful cases.This research also tested MPPI with some initial velocity and the MEDIRL-learned non-temporal Costmap with an extra velocity cost to maintain a target velocity at 10 m/s with the Mean Squared Error (MSE) cost between the target and current velocities. Although it showed a higher success rate compared to only using the original MEDIRL Costmap, it still reports a lot of failures. Finding an optimal cost function that weighs between the two costs, position, and velocity, is not an easy task, and even finding a good target velocity for accomplishing autonomous driving tasks is difficult. This reminds one of the main motivations of our spatiotemporal Costmap learning.

As expected, adding an extra safety check layer improved the success rates (88%) in all the models. However, failures happen even with the safety check layer when the collision checker does not determine that the collision would happen based on the other vehicle's velocity. Our future research will focus on improving the model to explicitly remove any potential collision-causing Costmap by itself, through a specific training procedure, so that it can achieve 100% success rate without any extra safety-checker.

7.4.2 Evaluation of Results

The same experiments are also conducted with a more realistic scenario by removing one of assumptions of having a near-perfect state estimation. It injects an additive White Gaussian noise with different variance $\Sigma_N = [0.1,0.1,0.02,1.0,1.0]$ for different states $[x, y, \psi, v, acc]$, where acc is the acceleration. The noise is added in the form of $c_s \cdot \epsilon$, with noise scale c_s and $\epsilon \sim \mathcal{N}$（$0, \sum$ N）, to the estimated state of the other vehicles.

As shown in Table 7-8, the performance degrades with bigger perception noise. From these experiments, it validates that there still exists a room for our method to improve, to make it more robust to real-world environments and to reduce the Sim2Real gap.

Table 7-8 Sensitivity analysis of medirl with perception noise

Noise scale	0.0	1.0	2.0	5.0
Success rate (%)	82	80	74	68

7.5 Summary

This chapter describes the numerical results of the collision avoidance of LC's decision-making and the success rate of LC's dynamic trajectory planning. The experiment results showed that the proposed methods can generate a series of actions to minimize the driving risk and prevent the host vehicle from collisions in both the scenarios with crowded static obstacles and the scenarios with dynamically moving vehicles. The MEDIRL algorithm can be improved where our Costmap can be directly used by MPC to accomplish a task without any hand-designing or hand-tuning of a cost function. And the proposed MEDIRL framework shows higher success rates in AVs' LC in a challenging dense traffic highway scenario in the CARLA.

Chapter 8. Summary and Conclusions

8.1 Introduction

This chapter will conclude the report with a summary of the simulation results. Direction for future work will also be provided.

8.2 Summary and Conclusions

8.2.1 Research of Decision Making based on RA-PRDQN

In this project, DRL algorithms combined with risk assessment functions are innovatively proposed to find an optimal driving strategy with the minimum expected risk. The experiment results showed that most of the proposed methods can generate a series of actions to minimize the driving risk and prevent the host vehicle from collisions in both the scenarios with crowded static obstacles and the scenarios with dynamically moving vehicles.

The most superior method among the examined ones is the RA prioritized replay DQN (RA-PRDQN) with the following features: 1) when the HV tends to drive out of the road boundary, the policy will correct the HV to return to the drive lane; 2) the agent will encourage the HV to drive at the center of the lane; 3) when the potential risk level is high, the strategy will generate a series of actions to reduce the risk level for potential near collision avoidance; 4) the agent can generate correct steering decisions for lane changing when a potential collision obstacle exists. Our proposed methods in this project could be further improved by mending the sampling probability function for vehicle position initialization, supplementing braking behavior for speed control, considering the influence of driving style, deploying continuous action space-based DRL algorithms, and optimizing the hyper-parameters in our proposed methods.

8.2.2 Research of Trajectory Planning based on MEDIRL

In this work, we showed a new cost function learning algorithm that improves the original MEDIRL algorithm where our Costmap could be directly used by MPC to accomplish a task without any hand-designing or hand-tuning of a cost function. Compared to the baseline methods, the proposed MEDIRL framework showed higher success rates in autonomous driving, lane keeping, and LC in a challenging dense traffic highway scenario in the CARLA simulator. We believe this work will serve as a steppingstone toward connecting IRL and MPC.

8.3 Directions for Future Research

8.3.1 Research of Decision Making based on PRDQN

As mentioned in Chapter 5, the mechanism of RA-PRDQN is a learning-free model which is dependent on the ground truth data. The most significant constraint of this is that the learning result could not transmit to any robust driving scenarios in the real world. The imitation learning-based model has been proposed to solve this problem, which could be applied to our research in the future.

The second problem is that the reward function estimated the risk level in this research focuses on safety control but not comfortability control. It is so significant to make sure a

comfortable driving environment for LC maneuver. So, our future research is to explore a reasonable reward function for PRDQN learning.

8.3.2 Research of Trajectory Planning based on MEDIRL

From the results and discussion mentioned in the last chapter, some failures happened even when the collision checker did not determine that the collision would happen based on the other vehicle's velocity. So, our future research will focus on improving our model to explicitly remove any potential collision-causing Costmap by itself, through a specific training procedure, so that it can achieve a 100% success rate without any extra safety-checker.

References

1. Ahmed, K. I., Ben-Akiva, M. E., Koutsopoulos, H. N., & Mishalani, R. G. (1996). Models of freeway lane changing and gap acceptance behavior. In Proceedings of the 13th international symposium on the theory of traffic flow and transportation.
2. Aleksandr, N., Vladimir, Z., & Anastasiya, F. (2018). Dynamic traffic re-routing as a method of reducing the congestion level of road network elements. Journal of Applied Engineering Science, 16(1). https://doi.org/10.5937/jaes16-15289
3. Alex, S., & Isaac, K. P. (2014). Traffic simulation model and its application for estimating saturation flow at signalised intersection. International Journal for Traffic and Transport Engineering, 4(3). https://doi.org/10.7708/ijtte.2014.4(3).06
4. Allen, M. (2020). Connected and networked driving: Smart mobility technologies, urban transportation systems, and big data-driven algorithmic decision-making. Contemporary Readings in Law and Social Justice, 12(2). https://doi.org/10.22381/CRLSJ12220209
5. Alperovich, T., & Sopasakis, A. (2008). Stochastic description of traffic flow. Journal of Statistical Physics, 133(6), 1083–1105. https://doi.org/10.1007/s10955-008-9652-6
6. Amodio, A., Panzani, G., & Savaresi, S. M. (2017a). Design of a lane change driver assistance system, with implementation and testing on motorbike. IEEE Intelligent Vehicles Symposium, Proceedings, 947–952. https://doi.org/10.1109/IVS.2017.7995837
7. Amodio, A., Panzani, G., & Savaresi, S. M. (2017b). Design of a lane change driver assistance system, with implementation and testing on motorbike. IEEE Intelligent Vehicles Symposium, Proceedings, 947–952. https://doi.org/10.1109/IVS.2017.7995837
8. Barlovic, R., Santen, L., Schadschneider, A., & Schreckenberg, M. (1998). Metastable states in cellular automata for traffic flow. European Physical Journal B, 5(3). https://doi.org/10.1007/s100510050504
9. Brackstone, M., & McDonald, M. (1999). Car-following: A historical review. In Transportation Research Part F: Traffic Psychology and Behaviour (Vol. 2, Issue 4). https://doi.org/10.1016/S1369-8478(00)00005-X
10. Caltagirone, L., Scheidegger, S., Svensson, L., & Wahde, M. (2017). Fast LIDAR-based road detection using fully convolutional neural networks. IEEE Intelligent Vehicles Symposium, Proceedings. https://doi.org/10.1109/IVS.2017.7995848
11. Chandler, R. E., Herman, R., & Montroll, E. W. (1958). Traffic Dynamics: Studies in Car Following. Operations Research, 6(2). https://doi.org/10.1287/opre.6.2.165
12. Chen, X., Sun, J., Ma, Z., Sun, J., & Zheng, Z. (2020). Investigating the long- and short-term driving characteristics and incorporating them into car-following models. Transportation Research Part C: Emerging Technologies, 117. https://doi.org/10.1016/j.trc.2020.102698
13. Choudhury, C. F., Email, T. F., Ben-akiva, M. E., & Rao, A. (2006). Modeling Cooperative Lane Changing and Forced Merging Behavior. Transport Science.
14. Chowdhury, D., Santen, L., Schadschneider, A., & Schadschneider, A. (2000). STATISTICAL PHYSICS OF VEHICULAR TRAFFIC AND SOME RELATED SYSTEMS Statistical physics of vehicular tra$c and some related systems. In Physics Reports (Vol. 329).

15. Coifman, B. (2003). Estimating density and lane inflow on a freeway segment. Transportation Research Part A: Policy and Practice, 37(8). https://doi.org/10.1016/S0965-8564(03)00025-9
16. Daganzo, C. F. (n.d.-a). A behavioral theory of multi-lane traffic flow. Part I: Long homogeneous freeway sections. www.elsevier.com/locate/trb
17. Daganzo, C. F. (n.d.-b). A behavioral theory of multi-lane traffic flow. Part II: Merges and the onset of congestion. www.elsevier.com/locate/trb
18. Daganzo, C. F. (1997). A CONTINUUM THEORY OF TRAFFIC DYNAMICS FOR FREEWAYS WITH SPECIAL LANES. In Transpn Res.-B (Vol. 31, Issue 2).
19. Daganzo, C. F. (2006). In traffic flow, cellular automata = kinematic waves. Transportation Research Part B: Methodological, 40(5), 396–403. https://doi.org/10.1016/j.trb.2005.05.004
20. Daganzo, C. F., & Cassidy, M. J. (2008). Effects of high occupancy vehicle lanes on freeway congestion. Transportation Research Part B: Methodological, 42(10), 861–872. https://doi.org/10.1016/j.trb.2008.03.002
21. Das, S., & Bowles, B. A. (1999). Simulations of highway chaos using fuzzy logic. Annual Conference of the North American Fuzzy Information Processing Society - NAFIPS. https://doi.org/10.1109/nafips.1999.781668
22. Dosovitskiy, A., Ros, G., Codevilla, F., López, A., & Koltun, V. (n.d.). CARLA: An Open Urban Driving Simulator.
23. Edie, L. C. (1963). Discussion of Traffic Stream Measurements and Definitions. 2nd International Symposium on the Theory of Traffic Flow.
24. Eidehall, A., Pohl, J., Gustafsson, F., & Ekmark, J. (2007). Toward autonomous collision avoidance by steering. IEEE Transactions on Intelligent Transportation Systems, 8(1), 84–94. https://doi.org/10.1109/TITS.2006.888606
25. Fagnant, D. J., & Kockelman, K. (2015). Preparing a nation for autonomous vehicles: Opportunities, barriers, and policy recommendations. Transportation Research Part A: Policy and Practice, 77, 167–181. https://doi.org/10.1016/j.tra.2015.04.003
26. Fu, X., Jiang, Y., Huang, D., Huang, K., & Wang, J. (2017). Trajectory planning for automated driving based on ordinal optimization. Tsinghua Science and Technology, 22(1). https://doi.org/10.1109/TST.2017.7830896
27. Gao, K., Yan, D., Yang, F., Xie, J., Liu, L., Du, R., & Xiong, N. (2019). Conditional artificial potential field-based autonomous vehicle safety control with interference of lane changing in mixed traffic scenario. Sensors (Switzerland), 19(19). https://doi.org/10.3390/s19194199
28. Gazis, D. C., Herman, R., & Weiss, G. H. (1962). Density Oscillations Between Lanes of a Multilane Highway. Operations Research, 10(5), 658–667. https://doi.org/10.1287/opre.10.5.658
29. Gipps, P. G. (1981). A behavioral car-following model for computer simulation. Transportation Research Part B, 15(2). https://doi.org/10.1016/0191-2615(81)90037-0
30. Gipps, P. G. (1986). A model for the structure of lane-changing decisions. Transportation Research Part B, 20(5). https://doi.org/10.1016/0191-2615(86)90012-3
31. Gora, P., Katrakazas, C., Drabicki, A., Islam, F., & Ostaszewski, P. (2020). Microscopic traffic simulation models for connected and automated vehicles (CAVs) - State-of-the-art. Procedia Computer Science, 170. https://doi.org/10.1016/j.procs.2020.03.091

32. Gu, X., Han, Y., & Yu, J. (2020). A novel lane-changing decision model for autonomous vehicles based on deep autoencoder network and XGBoost. IEEE Access, 8. https://doi.org/10.1109/ACCESS.2020.2964294
33. Gundaliya, P. J., Mathew, T. v., & Dhingra, S. L. (2008). Heterogeneous traffic flow modelling for an arterial using grid-based approach. Journal of Advanced Transportation, 42(4). https://doi.org/10.1002/atr.5670420404
34. Hata, A. Y., & Wolf, D. F. (2016). Feature Detection for Vehicle Localization in Urban Environments Using a Multilayer LIDAR. IEEE Transactions on Intelligent Transportation Systems, 17(2). https://doi.org/10.1109/TITS.2015.2477817
35. Hess, W., Kohler, D., Rapp, H., & Andor, D. (2016). Real-time loop closure in 2D LIDAR SLAM. Proceedings - IEEE International Conference on Robotics and Automation, 2016-June. https://doi.org/10.1109/ICRA.2016.7487258
36. Hidas, P. (n.d.). Modelling lane changing and merging in microscopic traffic simulation. www.elsevier.com/locate/trc
37. Hidas, P. (2005). Modelling vehicle interactions in microscopic simulation of merging and weaving. Transportation Research Part C: Emerging Technologies, 13(1). https://doi.org/10.1016/j.trc.2004.12.003
38. Holm, P., Tomich, D., Sloboden, J. &, & Lowrance, C. (2007). Traffic Analysis Toolbox Volume IV: Guidelines for Applying CORSIM Microsimulation Modeling Software. Report No. FHWA-HOP-07-079, IV(January).
39. Hoogendoorn, S. P., & Bovy, P. H. L. (2001). State-of-the-art of vehicular traffic flow modelling. Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering, 215(4). https://doi.org/10.1243/0959651011541120
40. IEEE ITSS, Institute of Transportation Engineers, & Institute of Electrical and Electronics Engineers. (n.d.). 2018 IEEE Intelligent Transportation Systems Conference: November 4-7, Maui, Hawaii.
41. Jiang, Y., Wang, S., Yao, Z., Zhao, B., & Wang, Y. (2021). A cellular automata model for mixed traffic flow considering the driving behavior of connected automated vehicle platoons. Physica A: Statistical Mechanics and Its Applications, 582. https://doi.org/10.1016/j.physa.2021.126262
42. Jin, L. S., Fang, W. P., Zhang, Y. N., Yang, S. bin, & Hou, H. J. (2009). Research on safety lane change model of driver assistant system on highway. IEEE Intelligent Vehicles Symposium, Proceedings. https://doi.org/10.1109/IVS.2009.5164426
43. Just, M. A., Keller, T. A., & Cynkar, J. (2008). A decrease in brain activation associated with driving when listening to someone speak. Brain Research, 1205. https://doi.org/10.1016/j.brainres.2007.12.075
44. Kang, K., & Rakha, H. A. (2018). Modeling driver merging behavior: A repeated game theoretical approach. Transportation Research Record, 2672(20). https://doi.org/10.1177/0361198118792982
45. Kesting, A., Treiber, M., & Helbing, D. (2007). General lane-changing model MOBIL for car-following models. Transportation Research Record, 1999. https://doi.org/10.3141/1999-10

46. Kiefer, R. J., & Hankey, J. M. (2008). Lane change behavior with a side blind zone alert system. Accident Analysis and Prevention, 40(2). https://doi.org/10.1016/j.aap.2007.09.018
47. Kim, H., Liu, B., Goh, C. Y., Lee, S., & Myung, H. (2017). Robust vehicle localization using entropy-weighted particle filter-based data fusion of vertical and road intensity information for a large-scale urban area. IEEE Robotics and Automation Letters, 2(3). https://doi.org/10.1109/LRA.2017.2673868
48. Knospe, W., Santen, L., Schadschneider, A., & Schreckenberg, M. (2004). Empirical test for cellular automaton models of traffic flow. Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics, 70(1). https://doi.org/10.1103/PhysRevE.70.016115
49. Kreusslein, M., Morgenstern, T., Petzoldt, T., Keinath, A., & Krems, J. F. (2020). Characterising mobile phone calls while driving on limited-access roads based on SHRP 2 naturalistic driving data. Transportation Research Part F: Traffic Psychology and Behaviour, 70. https://doi.org/10.1016/j.trf.2020.03.002
50. Li, B., Zhang, Y., Ge, Y., Shao, Z., & Li, P. (2017). Optimal control-based online motion planning for cooperative lane changes of connected and automated vehicles. IEEE International Conference on Intelligent Robots and Systems, 2017-September. https://doi.org/10.1109/IROS.2017.8206215
51. Lygeros, J., Godbole, D. N., & Sastry, S. (1998). Verified hybrid controllers for automated vehicles. IEEE Transactions on Automatic Control, 43(4). https://doi.org/10.1109/9.664155
52. Maerivoet, S., & de Moor, B. (2005). Cellular automata models of road traffic. In Physics Reports (Vol. 419, Issue 1). https://doi.org/10.1016/j.physrep.2005.08.005
53. McDonald, M., Wu, J., & Brackstone, M. (1997). Development of a fuzzy logic based microscopic motorway simulation model. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. https://doi.org/10.1109/itsc.1997.660454
54. Mendel, J. M. (1995). Fuzzy Logic Systems for Engineering: A Tutorial. Proceedings of the IEEE, 83(3). https://doi.org/10.1109/5.364485
55. Menendez, M., & Daganzo, C. F. (2007). Effects of HOV lanes on freeway bottlenecks. Transportation Research Part B: Methodological, 41(8). https://doi.org/10.1016/j.trb.2007.03.001
56. Milojevic, M., & Rakocevic, V. (2014). Distributed road traffic congestion quantification using cooperative VANETs. 2014 13th Annual Mediterranean Ad Hoc Networking Workshop, MED-HOC-NET 2014. https://doi.org/10.1109/MedHocNet.2014.6849125
57. Moridpour, S., Sarvi, M., Rose, G., & Mazloumi, E. (2012). Lane-changing decision model for heavy vehicle drivers. Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, 16(1). https://doi.org/10.1080/15472450.2012.639640
58. Nagatani, T. (1993). Self-organization and phase transition in traffic-flow model of a two-lane roadway. Journal of Physics A: General Physics, 26(17). https://doi.org/10.1088/0305-4470/26/17/005
59. Nagatani, T. (1994). Traffic Jam and Shock Formation in Stochastic Traffic-Flow Model of a Two-Lane Roadway. Journal of the Physical Society of Japan, 63(1). https://doi.org/10.1143/JPSJ.63.52

60. Nagel, K. (1996). Particle hopping models and traffic flow theory. Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics, 53(5). https://doi.org/10.1103/PhysRevE.53.4655
61. Nagel, K., & Schreckenberg, M. (1992). A cellular automaton model for freeway traffic. Journal de Physique I, 2(12). https://doi.org/10.1051/jp1:1992277
62. Nagel, K., Wolf, D. E., Wagner, P., & Simon, P. (1998). Two-lane traffic rules for cellular automata: A systematic approach. Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics, 58(2). https://doi.org/10.1103/PhysRevE.58.1425
63. Nie, L., Guan, J., Lu, C., & Zhou, M. (2018). Freeway Lane Changing Behavior Based on Fuzzy Logic. Beijing Gongye Daxue Xuebao/Journal of Beijing University of Technology, 44(3). https://doi.org/10.11936/bjutxb2017020001
64. Paz, A., Molano, V., Martinez, E., Gaviria, C., & Arteaga, C. (2015). Calibration of traffic flow models using a memetic algorithm. Transportation Research Part C: Emerging Technologies, 55. https://doi.org/10.1016/j.trc.2015.03.001
65. Pei, Y., & Xu, H. (2006). The control mechanism of lane changing in jam condition. Proceedings of the World Congress on Intelligent Control and Automation (WCICA), 2. https://doi.org/10.1109/WCICA.2006.1713670
66. Rickert, M., Nagel, K., Schreckenberg, M., & Latour, A. (1996). Two lane traffic simulations using cellular automata. Physica A: Statistical Mechanics and Its Applications, 231(4). https://doi.org/10.1016/0378-4371(95)00442-4
67. Rosique, F., Navarro, P. J., Fernández, C., & Padilla, A. (2019). A systematic review of perception system and simulators for autonomous vehicles research. In Sensors (Switzerland) (Vol. 19, Issue 3). https://doi.org/10.3390/s19030648
68. Samiee, S., Azadi, S., Kazemi, R., Eichberger, A., Rogic, B., & Semmer, M. (2016). Performance Evaluation of a Novel Vehicle Collision Avoidance Lane Change Algorithm. https://doi.org/10.1007/978-3-319-20855-8-9
69. Sarvi, M., & Kuwahara, M. (2007). Microsimulation of freeway ramp merging processes under congested traffic conditions. IEEE Transactions on Intelligent Transportation Systems, 8(3). https://doi.org/10.1109/TITS.2007.895305
70. Schadschneider, A. (2002). Traffic flow: A statistical physics point of view. Physica A: Statistical Mechanics and Its Applications, 313(1–2). https://doi.org/10.1016/S0378-4371(02)01036-1
71. Schadschneider, A., & Schreckenberg, M. (1993). Cellular automation models and traffic flow. Journal of Physics A: Mathematical and General, 26(15). https://doi.org/10.1088/0305-4470/26/15/011
72. Schakel, W., Knoop, V., & van Arem, B. (2012). Integrated lane change model with relaxation and synchronization. Transportation Research Record, 2316. https://doi.org/10.3141/2316-06
73. Schreckenberg, M., Barlović, R., Knospe, W., & Klüpfel, H. (2002). Statistical Physics of Cellular Automata Models for Traffic Flow. In Computational Statistical Physics. https://doi.org/10.1007/978-3-662-04804-7-7

74. Suh, J., Chae, H., & Yi, K. (2018). Stochastic Model-Predictive Control for Lane Change Decision of Automated Driving Vehicles. IEEE Transactions on Vehicular Technology, 67(6). https://doi.org/10.1109/TVT.2018.2804891
75. Thiemann, C., Treiber, M., & Kesting, A. (2008). Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data. Transportation Research Record, 2088, 90–101. https://doi.org/10.3141/2088-10
76. Toledo, T., Choudhury, C. F., & Ben-Akiva, M. E. (2005a). Lane-Changing Model with Explicit Target Lane Choice. Transportation Research Record: Journal of the Transportation Research Board, 1934(1). https://doi.org/10.1177/0361198105193400117
77. Toledo, T., Choudhury, C. F., & Ben-Akiva, M. E. (2005b). Lane-changing model with explicit target lane choice. Transportation Research Record, 1934. https://doi.org/10.3141/1934-17
78. Toledo, T., Koutsopoulos, H. N., & Ben-Akiva, M. (2007). Integrated driving behavior modeling. Transportation Research Part C: Emerging Technologies, 15(2). https://doi.org/10.1016/j.trc.2007.02.002
79. Toledo, T., Koutsopoulos, H. N., & Ben-Akiva, M. E. (2003). Modeling Integrated Lane-Changing Behavior. Transportation Research Record, 1857. https://doi.org/10.3141/1857-04
80. Toledo, T., & Zohar, D. (2007). Modeling duration of lane changes. Transportation Research Record, 1999. https://doi.org/10.3141/1999-08
81. Tomar, R. S., Verma, S., & Tomar, G. S. (2010). Prediction of lane change trajectories through neural network. Proceedings - 2010 International Conference on Computational Intelligence and Communication Networks, CICN 2010. https://doi.org/10.1109/CICN.2010.59
82. Ulbrich, S., & Maurer, M. (2015). Towards Tactical Lane Change Behavior Planning for Automated Vehicles. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 2015-October. https://doi.org/10.1109/ITSC.2015.165
83. von Neumann, J. (2017). The general and logical theory of automata. In Systems Research for Behavioral Science: A Sourcebook.
84. Wagner, P., Nagel, K., & Wolf, D. E. (1997). Realistic multi-lane traffic rules for cellular automata. Physica A: Statistical Mechanics and Its Applications, 234(3–4). https://doi.org/10.1016/S0378-4371(96)00308-1
85. Wahle, J., Neubert, L., Esser, J., & Schreckenberg, M. (2001). A cellular automaton traffic flow model for online simulation of traffic. Parallel Computing, 27(5). https://doi.org/10.1016/S0167-8191(00)00085-5
86. Wang, D., Hu, M., Wang, Y., Wang, J., Qin, H., & Bian, Y. (2016). Model predictive control-based cooperative lane change strategy for improving traffic flow. Advances in Mechanical Engineering, 8(2), 1–17. https://doi.org/10.1177/1687814016632992
87. Wang, Z., Zhao, X., Chen, Z., & Li, X. (2021). A dynamic cooperative lane-changing model for connected and autonomous vehicles with possible accelerations of a preceding vehicle. Expert Systems with Applications, 173. https://doi.org/10.1016/j.eswa.2021.114675
88. Wang, Z., Zhao, X., Xu, Z., Li, X., & Qu, X. (2021). Modeling and field experiments on autonomous vehicle lane changing with surrounding human-driven vehicles. Computer-Aided Civil and Infrastructure Engineering, 36(7). https://doi.org/10.1111/mice.12540

89. Wolcott, R. W., & Eustice, R. M. (2017). Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving. International Journal of Robotics Research, 36(3). https://doi.org/10.1177/0278364917696568
90. Wu, J., Brackstone, M., & McDonald, M. (2000). Fuzzy sets and systems for a motorway microscopic simulation model. Fuzzy Sets and Systems, 116(1). https://doi.org/10.1016/S0165-0114(99)00038-X
91. Wu, J., Brackstone, M., & McDonald, M. (2003). The validation of a microscopic simulation model: A methodological case study. Transportation Research Part C: Emerging Technologies, 11(6). https://doi.org/10.1016/j.trc.2003.05.001
92. Xia, Y., Qu, Z., Sun, Z., & Li, Z. (2021). A Human-Like Model to Understand Surrounding Vehicles' Lane Changing Intentions for Autonomous Driving. IEEE Transactions on Vehicular Technology, 70(5). https://doi.org/10.1109/TVT.2021.3073407
93. Xu, R., Huang, L., Yin, S., Yao, D., Zhang, H., & Peng, L. (2011). An analysis on traffic flow characteristics and lane changing behaviors in Beijing urban expressway bottlenecks. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. https://doi.org/10.1109/ITSC.2011.6082821
94. Yang, Q., & Koutsopoulos, H. N. (1996). A microscopic traffic simulator for evaluation of dynamic traffic management systems. Transportation Research Part C: Emerging Technologies, 4(3 PART C). https://doi.org/10.1016/S0968-090X(96)00006-X