UNIVERSITY OF CALIFORNIA
RIVERSIDE

A Personalized Behavior-Aware Motion Planning Framework for Intelligent Vehicle Operation

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

by

Xishun Liao

June 2023

Dissertation Committee:
Dr. Matthew J. Barth, Co-Chairperson
Dr. Guoyuan Wu, Co-Chairperson
Dr. Wei Ren

The Dissertation of Xishun Liao is approved:

_____

_____

Committee Co-Chairperson

_____

Committee Co-Chairperson

University of California, Riverside

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my two esteemed co-advisors, Dr. Guoyuan Wu and Dr. Matthew J. Barth. Their unwavering guidance, invaluable advice, and extensive knowledge were of utmost importance throughout this journey.

I want to express my special thanks to Dr. Guoyuan Wu for his meticulous guidance, patience, and dedication to this project over these past four years. His profound insights, unparalleled expertise, and unyielding enthusiasm for scientific research have been an inspirational beacon for me. They ignited my curiosity and continuously drove me to push forward, even during the most challenging moments. His support and encouragement have given me newfound confidence and ultimately allowed me to grow into a researcher.

I am profoundly grateful to Dr. Matthew J. Barth for his support and constructive suggestions. He has provided me with a platform that has granted me numerous invaluable opportunities. Because of this, I've been able to participate in major academic conferences, broaden my horizons, and connect with elite professionals from various fields.

I would also like to express my appreciation to Dr. Wei Ren for his contributions to my dissertation committee. I would also like to extend my gratitude to Dr. Ziran Wang for recommending me to the TSR group and for his mentorship during the initial stages of my PhD research. He has also provided me with guidance and support in planning my academic career. He has been not only a close mentor but also a great friend for many years. My thanks also go to another two research faculties in our lab, Dr. Peng Hao and Dr. Kanok Boriboonsomsin, for their managements and collaborations on the projects I involved in.

Furthermore, I would like to acknowledge my dearest friends, Hanzhi Zeng, Zhouqiao Zhao (and our feline companion Hugo), Hongming Chen, Haowen Zhu. Each in their own unique way, they have been by my side, providing comfort and companionship during this four-year journey. Also, I would like to express my sincerest thanks to my colleagues at TSR group, including Zhouqiao Zhao, Zhensong Wei, Xuanpeng Zhao, Jacqueline Garrido Escobar, Saswat Priyadarshi Nayak, Yejia Liao, Roland David Oswald, Abdullah Fuad Un-Noor, Zhengwei Bai, Haishan Liu, Dr. Ji Luo, Dr. Fei Ye, Dr. Nigel Williams, Dr. Chao Wang, Dongbo Peng, Chuheng Wei, Alexander Vu and many more.

Lastly, I would like to acknowledge the Department of Electrical and Computer Engineering (ECE) of UC Riverside, CE-CERT at UC Riverside, Honda R&D Americas, Inc., the California Department of Transportation, Toyota Motor North America, Volvo Group North America, United States Department of Transportation, and Esther F. Hays family for fully or partially supporting my study and research in this dissertation.

The major contents of the dissertation have been published in the Institute of Electrical and Electronics Engineers (IEEE) Transactions on Intelligent Transportation Systems, IEEE Transactions on Intelligent Vehicles, IEEE Transactions on Systems, Man, and Cybernetics: Systems, IEEE Internet of Things Journal, IEEE Transactions on Intelligent Vehicles, Transportation research Part D: Transport and environment, SAE International Journal of Connected and Automated Vehicles, IEEE Intelligent Transportation Systems Magazine, IEEE International Conference on Robotics and Automation, IEEE Intelligent Transportation Systems Conference, IEEE Intelligent Vehicles Symposium, IEEE Vehicular Technology Conference, and Transportation Research Board Annual Meeting.

DEDICATION

This dissertation is dedicated to my parents Liang Liao (廖亮) and Donghan Chen (陈冬寒) for their unconditional and endless love. With their support, I was able to explore the vast world. From going to Beijing for my undergraduate studies, to Maryland for my master's degree, and then to California for my doctorate, they were my solid backing. With their encouragement, I was able to bravely make each choice, break through the constraints of fate, and get through each difficult moment. They taught me what a family is and the meaning of family members. Without them, it would have been impossible for me to reach where I am today.

ABSTRACT OF THE DISSERTATION

A Personalized Behavior-Aware Motion Planning Framework for Intelligent Vehicles Operation

by

Xishun Liao

Doctor of Philosophy, Graduate Program in Electrical and Computer Engineering

University of California, Riverside, June 2023

Dr. Matthew J. Barth, Co-Chairperson
Dr. Guoyuan Wu, Co-Chairperson

The needs for personal travel and shipping freight keep increasing in our modern society. As a result, the size of our transportation networks is rapidly increasing, and the number of vehicles is also growing quickly. The resulting safety, mobility, and environmental sustainability problems are a major concern for the public. Intelligent Transportation Systems (ITS), utilizing Intelligent Vehicles (IVs) that possess automation capabilities and/or the ability to communicate with other agents, offer a promising solution. This ITS-

centric approach paves the way for traffic management without adding the cost of infrastructure expansion.

In the foreseeable future, IVs will be a critical component of ITS, and they are expected to coexist with conventional vehicles in mixed traffic scenarios. Such a setting demands an in-depth understanding of complex inter-vehicle interactions during motion planning. The potential for aggressive or overly cautious behaviors from IVs requires an awareness and prediction of other road users' behaviors.

To understand other road users, behavior modeling is a critical topic. An IV needs to model not only its own driver's behavior for a better user experience, but also other road users' behavior. With the development of sensing, communication, and cloud computing technologies, there will be more and more available data that can be shared. To take advantage of the abundant data, personalized behavior can be comprehensively studied to build a more accurate model. However, there is a concern on current "black box" strategies that could impede the acceptance and trust of IVs. Therefore, the need for developing data-driven, but explainable algorithms is critical.

This dissertation aims to develop a personalized behavior-aware motion planning framework for IV operation. The framework incorporates driver profile modeling, driving preference modeling, and interaction pattern recognition as part of behavior modeling. Furthermore, the motion planning component of the framework integrates the behavior model, thereby adapting to diverse driving styles and preferences in mixed traffic scenarios. The end goal is a safer, more efficient, and environmentally-friendly transportation system, aligning with the objectives of Intelligent Transportation Systems.

TABLE OF CONTENTS

LIST OF TABLES xi
LIST OF FIGURES xii
1 Introduction 1
1.1 Motivation 1
1.2 Research Objectives and Contributions 4
1.3 Dissertation Overview 5
2 Literature Review and Background 6
2.1 Review of Driving Behavior Modeling 6
2.1.1 Driver Profile Modeling 6
2.1.2 Data-Driven Vehicular Interaction Modeling 7
2.1.3 Personalized Driving Preference Modeling 8
2.2 Overview of Motion Planning 10
2.2.1 Prediction with Behavior Model for Motion Planning 12
2.2.2 Ramp Merging Planning Strategies for Intelligent Vehicles 13
2.3 Experiment Platforms 15
2.3.1 Co-Simulation Platform 15
2.3.2 Digital Twin Implementation for Intelligent Vehicles (IVs) 17
2.4 Challenges and Gaps 18
3 Overview of System Structure 21
3.1 Intelligent Vehicle Operation Pipeline 22
3.2 Integration of Behavior Modeling 24
4 Methodologies 26
4.1 Platform Construction 26
4.1.1 Vehicle-Edge-Cloud (VEC) Real-World Implementation Platform 26
4.1.2 Human-in-the-Loop Co-Simulation Platform 33
4.2 Personalized Driving Behavior Modeling 36
4.2.1 Driver Profile Modeling Based on Personality, Driving Style, and Mood States 36
4.2.2 Personalized Vehicular Interaction Pattern Analysis 57
4.2.3 Personalized Driving Behavior Modeling with Inverse Reinforcement Learning 76
4.3 Predictive Behavior-Aware Planning Strategy 87

4.3.1 Prediction for Driver Intention and Vehicle Trajectories 87
4.3.2 Ramp Merging Strategy with Feedforward Planning and Feedback Control 110
4.3.3 Driving Behavior Adaptive Advanced Driving Assistance System..... 140
5 Conclusions and Future Work..... 177
5.1 Conclusions..... 177
5.2 Related Publications to This Dissertation 179
5.3 Future Work and Discussion..... 180
BIBLIOGRAPHY..... 183

LIST OF TABLES

Table I Statistic Result of Communication Delay and Computation Time 31

Table II Example Fuzzy-Rules for Highway Scenario 43

Table III Personality Types-Driving Styles Distribution 52

Table IV Participants' Number of Driving Styles 52

Table V Prediction Result Evaluation 53

Table VI Permutation Feature Importance (PFI) Analysis 71

Table VII Weights of Recovered Cost Functions 84

Table VIII Prediction Accuracy Comparison in a 4-Second Prediction Window .. 108

Table IX Decision Table for Non-Cooperative Two-Person Game 123

Table X Decision Table for Cooperative Two-Person Game 124

Table XI Simulation Setup Parameters 129

Table XII Sensitivity Analysis for CAV Performance Measurement 137

Table XIII Training Result of NAR Neural Network 154

Table XIV Energy Consumption and Pollutant Emission Results of Human-in-the-Loop Simulation (All Values are on a Kilometer Basis) 158

Table XV Energy Consumption and Pollutant Emission Results 172

Table XVI Statistic Result of Communication Delay 175

LIST OF FIGURES

Fig. 1-1 Intelligent vehicle with automation and/or communication (adapted from [5])..... 2

Fig. 2-1 Schema of the motion planning components 11

Fig. 3-1 Personalized behavior-aware motion planning framework for IV at traffic system level 22

Fig. 3-2 Operation pipeline of intelligent vehicles with the integration of behavior modeling..... 23

Fig. 3-3 Schema of the proposed personalized behavior-aware motion planning framework for IV at vehicle level 25

Fig. 4-1 The vehicle-edge-cloud platform architecture validated in the field experiment 28

Fig. 4-2 Hardware in VEC platform: (a) NETGEAR hotspot (b) U-blox C94-M8P-2 GNSS unit, (c) Google Nexus 7, and (d) Dell R630 server..... 30

Fig. 4-3 Human-in-the-Loop Unity-SUMO co-simulation based on a real-world ramp merging area in Riverside, CA 34

Fig. 4-4 Bezier curve generator for CAV lane change maneuver..... 35

Fig. 4-5 Events design for driving style evaluation 40

Fig. 4-6 System workflow of the driver profile modeling (data collection and modeling) 42

Fig. 4-7 Correlation matrix between mood states and personality traits 47

Fig. 4-8 Personality type clustering dendrogram using HCA..... 48

Fig. 4-9 Average scores of three personality types..... 49

Fig. 4-10 Average mood states of three personality types 49

Fig. 4-11 Features in city driving session: (a) Maximum stop sign departure acceleration; (b) Maximum stop sign approaching deceleration; (c) Average speed over the limit (speed - limit); (d) Average minimum speed at a stop sign. 50

Fig. 4-12 Features in highway driving session: (a) Average speed over the speed limit; (b) Minimum time headway to slow vehicle; (c) Maximum brake force facing cut-in; (d) Lane change ratio of the session. 51

Fig. 4-13 Generalized vector autoregression under self-explaining neural network structure for Granger causality discovery 60

Fig. 4-14 Mapping object in Cartesian coordinate to Frenet coordinate. 64

Fig. 4-15 An example of estimating the conflict level of surroundings based on potential field theory 66

Fig. 4-16 On-ramp merging scenario from INTERACTION dataset. 69

Fig. 4-17 Interaction process during a four-vehicle merging. (a) Key moments during the merging process. (b) Interaction intensity among vehicles. 70

Fig. 4-18 Personalized interaction pattern analysis in ramp merging scenario 73

Fig. 4-19 Personalized driving preference modeling by recovering cost function. ... 78

Fig. 4-20 The urgency for a mandatory lane change. 81

Fig. 4-21 Personalized dataset collection using the Vehicle-Edge-Cloud digital twin platform 83

Fig. 4-22 Driving pattern comparison during a lane change process: (a) longitudinal speed of ramp vehicle (b) mandatory lane change urgency (c) lateral behavior and (d) fuel consumption. 85

Fig. 4-23 Personalized lane change behavior modeling: offline learning and online prediction 91

Fig. 4-24 Structure of the proposed neural network for lane-change decision prediction 92

Fig. 4-25 Polynomial trajectory generator 95

Fig. 4-26 Trajectory labeling based on DBSCAN 96

Fig. 4-27 Algorithm validation on UCR's human-in-the-loop co-simulation platform: predictions for lane change and most probable trajectory. 97

Fig. 4-28 Online lane-change prediction of a trip with one lane change event 98

Fig. 4-29 Online lane-change prediction of a trip without any lane change event 99

Fig. 4-30 The field implementation plan for lane change prediction. 101

Fig. 4-31 Vehicle HMI design. (a) Speed guidance, and (b) detection for lane changing prediction 102

Fig. 4-32 The bird-eye's view application interface: digital twin of vehicles running on the edge server in real time. 103

Fig. 4-33 Field implementation for online lane change prediction with visualization on camera and edge server: (a)(b)(c) Prediction for Driver 1, changing the lane behind MV1. (d)(e)(f) Prediction for Driver 2, changing the lane in front of MV1.104

Fig. 4-34 Prediction result analysis of using personalized models..... 107

Fig. 4-35 System workflow of the mixed traffic ramp merging strategy for CAVs 114

Fig. 4-36 Conflict prediction module..... 115

Fig. 4-37 Hysteresis controller for disturbance filtering. 125

Fig. 4-38. Unity-SUMO integrated simulation based on a real-world ramp merging area in Riverside, CA: (a) View from Google Maps at the real-world ramp; (b) User interface of the Unity-SUMO co-simulation platform; (c) A CAV (in red) with radar system and a legacy vehicle (in white). 127

Fig. 4-39 Game process of a non-cooperative game: (a) An on-ramp CAV and a mainline legacy vehicle compete for merging; (b) Merging order is determined; (c) the whole process of the game. 130

Fig. 4-40 Game process of a cooperative game: (a) Two CAVs compete for merging; (b) Merging order is determined; (c) the whole process of the game. 131

Fig. 4-41 Speed of traffic flows: (a) Average speed of mainline vehicles; (b) Average speed of on-ramp vehicles. 133

Fig. 4-42 Average fuel consumption: (a) Fuel consumption of mainline vehicles; (b) Fuel consumption of on-ramp vehicles. 134

Fig. 4-43. Average driving volatility: (a) Speed volatility of mainline vehicles; (b) Speed volatility of on-ramp vehicles..... 136

Fig. 4-44 Weight selection for sensitivity analysis..... 138

Fig. 4-45 Sensitivity analysis for CAV performance measurement: (a) Average fuel consumption, (b) average speed, and (c) average acceleration volatility. 139

Fig. 4-46 System architecture of the proposed learning-based driver behavior modeling system 144

Fig. 4-47 Human machine interface of speed suggestion in simulation and real-world vehicle..... 145

Fig. 4-48 One of the cooperative merging scenario at on-ramp built in Unity..... 146

Fig. 4-49 HCA cluster visualization in 3-D space..... 150

Fig. 4-50 Structure of NAR neural network..... 153

Fig. 4-51 Speed error comparison in the game engine-based simulation 156

Fig. 4-52 V2C-based cooperative merging scenario at on-ramp 160

Fig. 4-53 General architecture of the real-world cooperative ramp merging ASA system..... 162

Fig. 4-54 Merging scenarios: (a) merging into a vehicle string, (b) merging after a vehicle or string, (c) merging in front of a vehicle or string, and (d) merging to an empty mainline..... 163

Fig. 4-55 Perspectives of (a) ramp vehicle driver and (b) mainline vehicle 2 driver while conducting the cooperative ramp merging..... 165

Fig. 4-56 V2C-based cooperative merging stages..... 166

Fig. 4-57 Digital Twin of vehicles running on the cloud server in real time..... 169

Fig. 4-58 Cloud-based cooperative merging scenario at on-ramp..... 170

Fig. 4-59 Communication delay results: (a) Time series of communication delay and (b) Distribution of communication delay under its 99th percentiles..... 174

1 Introduction

1.1 Motivation

Traffic related issues such as safety, efficiency, and environmental sustainability have drawn significant attention as transportation is becoming more involved in people's daily lives. Stated by Hedges & Company in 2021[1], there were 1.446 billion motor vehicles in-use in the world, and this number will likely grow to two billion within one or two decades. Furthermore, according to the latest data from the National Highway Traffic Safety Administration (NHTSA)[2], an estimated 42,939 people died in motor vehicle traffic crashes in 2021, a 10.1% increase compared to 39,007 fatalities reported in 2020. A survey from INRIX showed that traffic congestion cost each American nearly 100 hours of lost time at a cost of $1,400 in 2019, an increase of compared to 2017 [3]. Traffic congestion also caused 44.3 billion liters of fuel to be wasted worldwide in 2015, according to U.S. Department of Energy [4]. With these concerns, it is clear we need to advance our current transportation systems.

As a potential solution, the development of intelligent transportation system solutions can enhance traffic management, thereby improving safety, mobility, and reliability without the cost of additional infrastructure. Within the field of intelligent transportation systems, there has been significant growth in intelligent vehicles (IVs) due to advancements in perception, communication, and computation technologies. IVs such as automated vehicles, connected vehicles, and connected and automated vehicles are those that have certain automation capability and/or can communicate with other entities, as shown in Fig. 1-1.

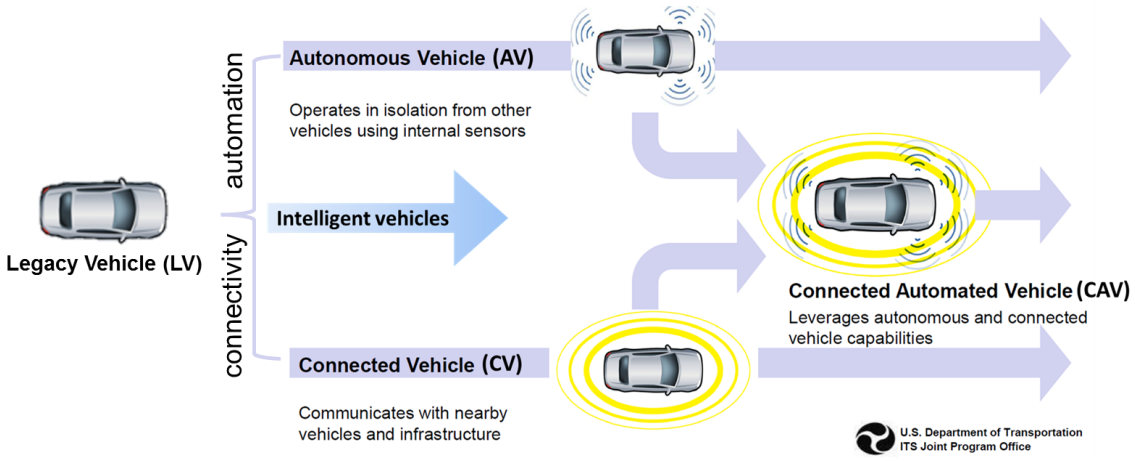


Fig. 1-1 Intelligent vehicle with automation and/or communication (adapted from [5])

Autonomous vehicle (AV) technology consists of sensing, perception, planning and control technologies, allowing operation of the vehicle without direct driver input to control the steering, acceleration, or braking. Moreover, the level of automation varies from a human-driven vehicle to fully automated vehicle. Defined by SAE International in their J3016 [6], six levels of automation are: 0) no automation, 1) driver assistance (longitudinal or lateral vehicle motion control), 2) partial driving automation (longitudinal and lateral vehicle motion control), 3) conditional driving automation, 4) high driving automation, and 5) full Driving Automation.

In addition to automated vehicle technology, connected vehicle (CV) technology further strengthens intelligent vehicles, which takes advantage of vehicle-to-everything (V2X) communication, allowing vehicles to communicate with other road participants and hence conduct more efficient maneuvers. V2X-enabled connected vehicles are equipped with a cellular network and/or Dedicated Short-Range Communications (DSRC) [7], [8], allowing for communicating with each other. Representative V2X scenarios include

vehicle-to-vehicle (V2V) communication, vehicle-to-infrastructure (V2I) communication, and vehicle-to-cloud (V2C) communication. The communication module of a connected vehicle can provide additional information that cannot be readily detected by perception sensors (if any) and can generally provide information more quickly than through sensor detection and processing [9]. Combining automation and connectivity, connected and automated vehicles (CAV) can better utilize shared information and optimize system-wide performance.

However, current intelligent vehicle systems face several obstacles that impede their widespread adoption and effectiveness. Drivers' preferences and abilities vary greatly, necessitating personalized approaches to motion planning that cater to individual styles and behavior patterns. While automation may offer optimal solutions, it must also be cognizant of drivers' preferences, as the acceptance and trust in the black box strategy remain weak among users. Moreover, interacting with human drivers poses a significant challenge for intelligent vehicles, as the uncertainties surrounding human behavior make it difficult to predict and respond appropriately in real-time situations.

Considering these pressing concerns, this dissertation aims to present a personalized behavior-aware motion planning framework for intelligent vehicle operation. By integrating behavior-awareness into motion planning algorithms, the framework can understand the behavior of human beings and further provide optimal and favorable operation strategies that are beneficial to the whole traffic system.

1.2 Research Objectives and Contributions

Studies with respect to IVs (i.e., AVs, CVs, and CAVs) have been conducted for many years, and the operation process is commonly divided into sensor fusion, perception, motion planning, and control [10]. Among the IV's operation process, this research focuses on the motion planning part. The primary objectives of this research are as follows:

- Modeling personalized driving behavior: This research aims to integrate the human factor into the motion planning framework for intelligent vehicle (IV) operation. By considering drivers' preferences, abilities, and behavior patterns, this research seeks to develop a comprehensive understanding of personalized driving behavior.
- Development of human-centric motion planning: A secondary objective of this research is to utilize the developed understanding of human behavior and preference in the motion planning phase to create a human-centric operational strategy. This strategy should account for the variable nature of human driving behavior and adapt IV operations to best mimic or cooperate with human road users. This objective will ultimately facilitate an increased level of reliability and favorability in IV operation, enhancing the real-world applicability and acceptance of IVs.

The specific contributions of this dissertation can be summarized as follows:

- A systematic framework has been developed to study personalized driving behavior, accounting for unique elements such as driver profiles, driving preferences, and implicit interaction patterns.

- A novel behavior-aware motion planning strategy for IVs has been proposed for mixed traffic environments.
- A vehicle-edge-cloud digital twin platform for real-world implementation and a human-in-the-loop co-simulation platform have been devised, enabling personalized driving behavior dataset generation, algorithm development and validation.

1.3 Dissertation Overview

The rest of this dissertation is organized as follows: Chapter 2 provides a literature review and background study on driving behavior modeling, motion planning, and experiment platforms, offering a broad understanding of the field. Following this, Chapter 3 gives an overview of the system structure, discussing the Intelligent Vehicle operation pipeline and the integration of behavior modeling. Chapter 4 elaborates on the methodologies employed during the research, including platform construction, personalized driving behavior modeling, and the development of a predictive behavior-aware planning strategy. Lastly, Chapter 5 wraps up the report with a conclusion summarizing the research findings and potential future work to fulfill the research objectives. It also includes a list of publications related to this dissertation.

2 Literature Review and Background

2.1 Review of Driving Behavior Modeling

2.1.1 Driver Profile Modeling

Driver profiles are critical to improving the understanding of driver preferences, which could lead to providing personalized products with better suggestions (e.g., advanced driver assistant system) [11], boosting user acceptance and trust of AV via shared personality [12], and predicting the behaviors of other drivers [13].

A driver profile is determined by the demographic, physiological, and behavioral characteristics of a driver [14]. The majority of driver profile studies focused on behavioral characteristics, personality traits, mood states, and driving style are the three most common research subjects. Personality traits usually refer to individual differences in characteristic patterns of thinking, feeling, and behaving [15]. Meanwhile, mood states are defined as emotional state that affects the way people respond to stimuli [16], but unlike mood states and personality traits, there is no agreed-upon definition for driving style [17]. Modeling behavioral characteristics is more complex as they are associated with a variety of temporal factors (e.g., traffic condition, surrounding vehicles, weather, and time of the day) [18]. Additionally, the lack of a consistent and controlled environment for data collection may result in difficulty modeling and validating the behavioral characteristics. This led to the popularity of survey-based studies as a method of choice for recording behavioral characteristics. Survey-based methods are popular as the process of their experiments are stable and have higher replicability. Among young drivers, Wu et al. [19] found a correlation between personality traits and driving styles whilst Zimasa et al. [20] related

negative mood states to dangerous driving. Garrity and Demick [21] revealed the relationship among personality traits, mood states, and driving style pairwise but the driving style evaluation of this study may be biased as it was evaluated subjectively by the passengers.

Survey-based approaches may introduce bias to driving style recognition as drivers may not perceive their performance correctly [22]. In the past, studies defined driving style based on research goals with objective methods, which utilize selected vehicle operation states (e.g., speed, acceleration, and angular speed) from the naturalistic driving dataset [23], driving simulators [24], or test vehicles [25]. In addition to the mode of data collection, drivers were classified from mild to aggressive [26], driving performance from bad to good [27], and by analyzing dynamic demand i.e., sports, moderate, and economical driver [25].

While driver profiles have been analyzed in previous studies, few researchers have been able to systematically research driving style, mood states, and personality traits in combination to build driver profiles. As mentioned earlier, past studies assessed the driving style subjectively based on surveys or identified it based merely on vehicle operation states in an inconsistent environment, which changes from time to time due to uncontrolled temporal factors.

2.1.2 Data-Driven Vehicular Interaction Modeling

Many vehicle interaction studies have been applied to extract interaction patterns from observed vehicle trajectories. Considering limited computational resources in real-time, Refaat et al. [28] prioritized surrounding agents for the planning process, based on their

influence on the ego vehicle. Also, the attention mechanism is a popular interaction encoding method as it can evaluate the similarity or importance between vectorized series [29]. To quantify the influence of others on ego vehicle, Leurent and Mercat [30] calculated the attention value between ego vehicle and its surrounding agents and hence improved the prediction result. In addition to analyzing such influence, interaction study considers the mutual effect. Researchers [31]–[33] modeled the vehicle interaction using graph neural network (GNN), which can capture the correlation between nodes (i.e., vehicles) and encode the interaction intensity in the learnable weights of graph edges. To capture both the influence and interaction, graph attention networks (GAT) were adopted to update the vehicle features by considering the influence from other nodes [34]. However, supervised or semi-supervised methods are limited by data labeling cost, and GNN is not flexible in dynamic environments because the graph cannot handle large variations in the number of nodes (i.e., agents) [35]. Other researchers explored parametric methods such as social value orientation (SVO) [36], [37] and inverse reinforcement learning (IRL) [38], [39] for understanding drivers' interaction preference. Although these models are explainable, they learn fixed parameters by using historical data and cannot update in real time. Moreover, the aforementioned methods cannot explicitly explain the interaction by addressing three key questions related to ' who ', ' when ', and ' how '.

2.1.3 Personalized Driving Preference Modeling

A personalized driver model is usually learned based on the dataset from a specific driver (i.e., learning from demonstration) and is mainly used for prediction, planning, and control [40]. Drivers’ preferences in their vehicle states are well-studied. A personalized driving

assistant system developed by Lefevre et al. could identify the current driving maneuver and predict the steering and acceleration, facilitating control assistance [41]. Considering the occupants' preference for lateral and longitudinal comfort, Bae et al. proposed a personalized control system enabling autonomous vehicles to drive like human beings [42]. Also, algorithms based on Gaussian Mixture Model and HMM are popular for prediction of personalized driving behavior and are deployed in a variety of applications such as personalized lane departure prediction [43] and personalized car-following behavior prediction [44].

Not only the vehicle states but also the surroundings and vehicular interaction should be considered in drivers' preference. To integrate the interaction into behavior modeling, Huang et al. included the awareness of the effect of the ego vehicle on the surrounding vehicles into the cost function [45]. The driver's preference over vehicle states and interactions can be expressed by the cost function recovered by inverse reinforcement learning (IRL), which assumed that human behavior was motivated by optimizing the unknown reward function [46]. In [47], the interaction behavior under different conditions was formulated as a cost function with different linear combinations of features and learned by continuous IRL. The cost function with interpretable weights on features opens the black box of behavior modeling, by showing the diversity of feature attention in different scenarios. Furthermore, the cost function can be adjusted as needed to reflect changing driving conditions and can always provide a solution aligning with driver preference in unseen scenarios.

Collecting a personalized dataset for interaction study is another major limitation in interaction modeling and driving personalization study. The difficulty of scenario reproduction constrains the driver's experiences in similar interactive conditions and the collection of enough data for model training.

2.2 Overview of Motion Planning

Defining the motion planning process is complex because the relationship of planning and decision making is highly coupled. Katrakazas et al. defined the planning as a three step process: 1) finding a path, 2) searching for the safest maneuver, and 3) determining the most feasible trajectory [48]. In this definition, path planning as the first step, is responsible for finding geometric feasible paths from an initial state to a given terminating state. Then maneuver planning (decision making) selects the desired lane and velocity profile, based on driver's intention and both current and predicted information (e.g., 'going straight', 'turning', 'overtaking' etc.). Trajectory planning aims to generate suitable speed profiles. Some researchers prefer to combine path planning and maneuver planning together [10], [49]. For example, Karl et al. proposed a joint maneuver planning and motion planning method, which is not a stand-alone component for solving the motion-planning problem, since the model-predictive control (MPC) typically requires a precomputed reference trajectory [50]. A more general definition was proposed by Claussmann et al., and the maneuver planning and motion planning process is to select one sequence among the generated motions, which are sequences of paths, trajectories, maneuvers, or actions [51]. The path planning module and maneuver planning module form a closed loop to find

the best maneuver with an optimal path for the vehicle to follow, and then the motion planning module is responsible for achieving the selected path with the best trajectory.

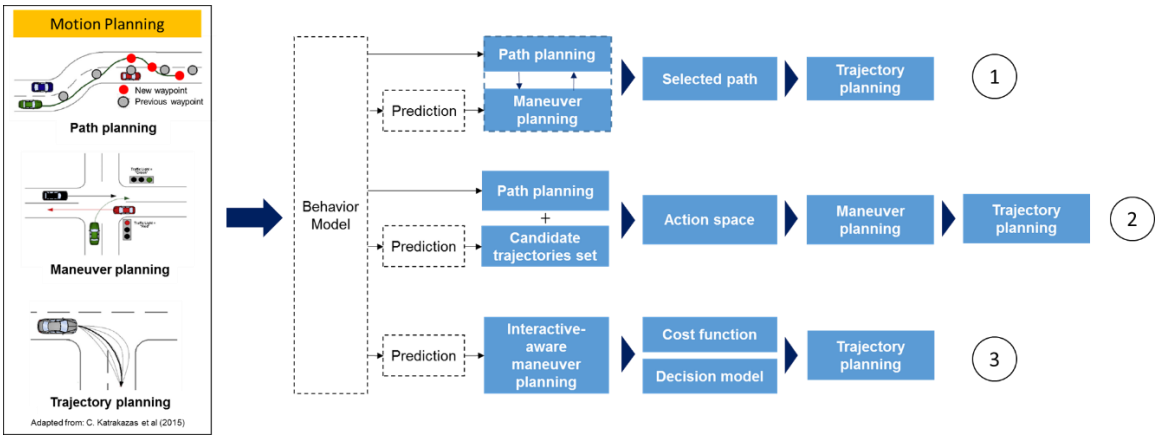


Fig. 2-1 Schema of the motion planning components

A schema overview is shown as Fig. 2-1, which distinguishes the planning and decision making into three approaches. In the first two sequential approaches are the commonly used, where the behavior layer is an optional component serving as a preference interpreter. In approach 1, the path planning module and maneuver decision module form a closed loop to find the best maneuver with an optimal path for the vehicle to follow, and then the motion planning module is responsible for achieving the selected path with the best trajectory. The difference between approach and approach 1 is the order of each module. When the consequence of each action or the feasible action is unsure, approach 2 searches the action space first and then decides which maneuver is the best.

When researchers consider the interaction into motion planning, approach 3 is adopted to integrate the behavior layer into maneuver planning and trajectory planning, providing a maneuver planning model for the next level control. For example, an objective function can be used in MPC and reinforcement learning, and the decision model, such as the game

theory decision tree that consists of discrete actions that each agent, can provide the best action at each time step. In conclusion, these three approaches take the input from sensor and communication and provide suitable trajectories for the control module.

2.2.1 Prediction with Behavior Model for Motion Planning

Prediction is crucial for motion planning because it allows an autonomous system to anticipate and react to future events or changes in the environment. By predicting the future state of objects, obstacles, and agents, the system can proactively plan its own actions to avoid collisions, achieve its goals, and navigate safely and efficiently. Further, combining a driver behavior model with prediction enhances the motion planning process by incorporating human-like driving strategies and behaviors, further improving the vehicle's ability to navigate complex scenarios.

Therefore, there is an increasing amount of literature that recognizes the importance of combining the driving behaviors model into the prediction. Hidden Markov Model (HMM) was widely used to infer the lane-change intention [52]–[54] and is usually integrated with the Bayesian network [55] to recognize the lane-change behavior. As lane-change intention prediction can be modeled as a classified problem, the multilayer perceptron (MLP) was used as a discriminator [56] in long-term lane-change prediction. Moreover, deep learning methods, such as Long Short-Term Memory (LSTM) model, achieved a precision of 90.5% on time-series problems [57]. To find out relevant features for lane changing in a time series, Scheel et al. [58] integrated a temporal attention mechanism with LSTM to improve the prediction accuracy to 92.6% and provide understandability on feature importance. Besides lane change intention prediction, lane change trajectory prediction is also a critical

problem. Based on the beam search technique, Park et al. [59] adopted Seq2seq LSTM to produce K most likely trajectories on an occupancy grid map. By adding information on traffic level and vehicle types, Xue et al. [60] adopted XGBoost for lane change decision prediction and LSTM for trajectory prediction, achieving a Mean Square Error of 6.62m for trajectory prediction.

However, most of the algorithms ignored vehicular interaction with the surroundings. Furthermore, supervised learning methods were limited by the lack of enough labeled datasets to cover each possible scenario, and very little online validation was carried out as well as real-world validation.

2.2.2 Ramp Merging Planning Strategies for Intelligent Vehicles

A number of ramp merging strategies have been developed to increase road safety and efficiency by leveraging CAV technology [9]. Awal et al. proposed a proactive optimal merging strategy based on V2V communication to optimize the on-ramp merging time and to reduce merging bottlenecks [61]. Utilizing V2I communication, Scarinci et al. developed a cooperative merging assistant control system based on the combination of macroscopic and microscopic traffic flow theories to create gaps to allow on-ramp vehicles to merge [62]. Lu et al. introduced a concept of virtual platooning and developed a closed-loop adaptive longitudinal control algorithm to control the merging speeds of CAVs [63]. Jain et al. proposed an adaptive strategy to platoon merging with vehicle engine uncertainty, and by considering the bidirectional error, the merging vehicle can interact with both front and rear vehicles [64].

Apart from the aforementioned methodologies, game theory has also been widely adopted in ramp merging strategies for CAVs. Some researchers adopted game theory for decision-making in a complex environment. To get a global perspective and obtain the optimal solution, centralized optimization algorithms have been developed to coordinate the ramp merging maneuvers. Jing et al. designed a cooperative game-based merging sequence coordination system to arrange CAVs into platoons, and used optimal control to guarantee the best sequence in terms of mobility and fuel consumption [65]. Ramp merging can be seen as a mandatory lane change behavior, and many studies for mandatory lane change can be adopted to ramp merging management. To mitigate shockwaves caused by merging maneuvers, Akti et al. proposed a game theory-based algorithm to organize the longitudinal and lateral movements for merging vehicles, in a fully connected environment [66]. Wang et al. combined receding horizon control with game theory to find an optimal acceleration control for both lane-changing and car-following [67]. Based on a large amount of real-world vehicle trajectories, the game theory-based algorithm becomes more powerful with the support of human behavior estimation. By estimating surrounding vehicles' aggressiveness as their utilities, Zhang et al. presented a game theory-based model predictive controller to find out the optimal gap to perform mandatory lane-changing, by searching up to three gaps on the adjacent lane [68].

However, the majority of these studies rely on a strong assumption of 100% CAV penetration rate, allowing for a centralized complete game approach that can utilize full information [69]. In contrast, especially in mixed traffic with low penetration rate, CAVs can only form an incomplete game with limited information from the legacy vehicles

within the detection range of CAVs. Moreover, the advantage of CAVs' long-distance communication is diminished in the mixed traffic environment since long-distance communication includes higher uncertainty of the environment.

2.3 Experiment Platforms

2.3.1 Co-Simulation Platform

Before modeling personalized driving behavior in the real world, algorithm development is usually performed in simulation. Simulation is a widely used method to implement and evaluate algorithms due to its cost-effective, risk-free, and interactive characteristics. Microscopic traffic simulators, such as PTV VISSIM [70], Aimsun [71] ,and SUMO[72], provide high fidelity and continuous traffic simulation to model complex vehicle interactions in a specific traffic network [73]. In recent years, game engine-based driving simulators, such as LGSVL (Unity-based) and CARLA (Unreal-based) [74], have gained much momentum and attracted considerable attraction from researchers. They can simulate more realistic scenarios with a high degree of freedom, which enables the use of various vehicle models, customization of sensor suites and ambient environment, full control of all static and dynamic actors, and map generation.

Currently, each individual simulator has its own advantages and focus arenas. However, a single simulator is not enough for modeling and evaluating cooperative automated driving system (CADS) design as well as establishing a realistic testing environment. Some recent research integrated multiple simulators to leverage their capabilities. Oh et al. built a virtual reality (VR)-in-the-loop simulator by connecting VISSIM and Unity with VR and created an immersive driving environment [75]. To further explore the vehicle-to-anything

(V2X) communication in large-scale traffic, Jia et al. integrated three popular open-source simulators SUMO (traffic simulator), OMNeT++ (network simulator), and Webots (3D robot simulator), providing the information of large-scale network in addition to traffic simulation [76]. However, compared to the aforementioned game engine-based simulators, Webots in this work lacks a high-fidelity simulation environment and vehicle models. Biurrun-Quel et al. configured a driver-centric simulator by connecting Unity and SUMO through TraCI (Traffic Control Interface) protocol to present a driver view for one of the SUMO controlled background vehicles [77], which only allows one-way communication from SUMO to Unity. To better simulate the mixed traffic environment and implement the algorithm for CAVs, two-way communication needs to be set up.

Furthermore, human-in-the-loop (HuiL) is a prototype platform for quickly exploring novel in-the-loop applications that can enhance the interactions between human beings and the physical world [78]. HuiL is widely used in different research topics highly related to human interaction with control systems. For example, in the work of rollover prevention for sport utility vehicles, the researchers validated the performance of the anti-rollover control via HuiL [79]. With HuiL, Li et al. [80] proposed a synthetic approach to solving safety-critical interaction problems in the SAE Level 3 automated vehicles which are mostly autonomous and only need limited driver intervention. Szalai et al. [81] created a mixed reality simulation environment allowing real-world vehicles to interact with virtual traffic flow generated by SUMO and to be visualized in Unity. It collected the human driving behavior information with real-world vehicles' sensors to serve as a validation procedure for autonomous vehicle development.

2.3.2 Digital Twin Implementation for Intelligent Vehicles (IVs)

The recent emergence of digital twin technology has attracted a significant amount of attention from both academia and industry. The global digital twin market size was reported to be valued at $5 billion in 2020, and will be expanded to $86 billion in 2028, with a compound annual growth rate of 43%. Among all end-users like manufacturing, energy, and health care, the automotive and transportation industry took one of the largest shares in the global digital twin market in 2020 [82].

By a widely adopted definition (with some variations), a digital twin is a digital replica of a living or non-living physical entity [83]. This concept got to be known by most people in the early 2010s, when NASA adopted it as a key element in its technology roadmap [84]. During the past few years, digital twins have been applied to different vehicular systems. Particularly, Chen et al. developed a "Driver Behavior Twin" to allow driver behavioral models to be shared among multiple connected vehicles to predict future actions of surrounding vehicles [85]. Although this study did not come up with a solid network architecture, its concept did inspire a series of subsequent studies by this dissertation.

In 2020, the authors first proposed a digital twin paradigm for advanced driver-assistance systems (ADAS) with a cloud architecture, which enables the communication between real vehicles and their digital twins deployed on the cloud server in real time [86]. This cooperative ramp merging ADAS was later validated in a field implementation with real passenger vehicles and a private cloud server at University of California, Riverside [87]. Later, the authors introduced edge computing to this network architecture by proposing a mobility digital twin framework, which includes not only vehicle digital twins but also

human and infrastructure digital twins [83]. Some of the detailed aspects of digital twins for CAVs have also been studied by the authors, such as how to visualize the digital twin information [88], how to leverage the digital twin information for cooperative driving scenarios [89], and how to build a simulation environment to model digital twins [90]. Many of the aforementioned studies have been summarized in a survey paper, where the role of digital twins in CAVs is also compared with the roles of several similar technologies, such as iteration, model-based design, and parallel driving [91].

2.4 Challenges and Gaps

To address the traffic problems of safety, mobility and energy consumption, the module of decision and planning always serves as a key component. Schwarting et al. [10] pointed out several questions to be answered: 1) What informs the decision-making process of vehicles regarding their next maneuvers? 2) How do vehicles utilize sensor-gathered data in making immediate and future decisions? 3) How does the situation and behavior of other vehicles influence a vehicle?4) How can vehicles leverage their past experiences and human driving behaviors to enhance their driving skills? 5) What measures can be implemented to ensure the reliability and safety of vehicle control and planning systems? And 6) What strategies can be used to effectively coordinate multiple vehicles on the road simultaneously? Following these questions, we may lay the groundwork for motion planning. Nevertheless, there remain substantial challenges. Our traffic systems are far from ideal and uniform, encompassing various types of vehicles, drivers, automation levels, connectivity capabilities, and diverse human thought processes.

The first challenge is the “mixed” traffic conditions. In the near future, the road is anticipated to accommodate different types of vehicles. For instance, Intelligent Vehicles (IVs) will coexist with traditional vehicles, and among IVs, capabilities and algorithms will differ. While many studies on IVs assume a full penetration rate, it is more pragmatic but challenging to consider interactions between different vehicle types in such mixed traffic situations.

Similarly, the second challenge is the diversity of drivers. Human behavior must be considered due to the vast range of driving styles. Although there has been significant research on driver categorization (such as aggressive, normal, cautious), such broad classifications are inadequate for effective motion planning. Human beings display varying behavior in different situations. Although human drivers are typically assumed to be rational, they can exhibit irrational behavior in rare circumstances; for example, driver mistakes can be made due to misjudgments or distraction leading to erratic or unpredictable driving. In some abnormal traffic conditions (e.g., sudden traffic jams and extreme weather), drivers might react unsafely. In these instances, it is important to study personalized driver behavior and abnormal situations.

Moreover, it is critical to incorporate real-world engineering constraints into these studies. This represents not only a highly practical approach but also a pressing demand, marked as the third challenge. Many researchers focus solely on enhancing IV operation as a universal remedy for all complex issues. However, given the complexity of the field, this approach proves to be a daunting task. With standardization (such as communication protocols) still posing a challenge, IVs require support from both physical and digital (e.g.,

cloud service) infrastructure. The proposed algorithm needs to be flexible, reliable, and scalable, allowing optimization based on different resource constraints.

Lastly, it is crucial to design vehicles capable of effective human interaction, maintaining a human-centric design approach. This demands a thoughtful consideration of the equilibrium between personalized and collective behavior.

To conclude, the research gaps of Intelligent Vehicle (IV) research include:

- Current automation technologies might be optimal but not necessarily favorable or acceptable to all users;
- The preferences and abilities of drivers vary significantly, which can complicate the design and implementation of IV systems;
- There is a weak acceptance of and trust in “black box” strategies due to their lack of transparency; and
- Interaction with human beings poses a significant challenge for IVs due to the unpredictability and complexity of human behaviors.

3 Overview of System Structure

The general architecture of the proposed strategy for IV operation is illustrated in Fig. 3-1, which consists of a physical world and a digital world. In the physical world, there are four types of vehicles, Legacy Vehicles or human driven vehicles (LVs or HDVs), Connected Vehicles (CVs), Autonomous Vehicles (AVs), and Connected and Automated Vehicles (CAVs), all forming mixed traffic. Due to their communication capability, CVs and CAVs can share a database and collaborate, making joint decisions through cloud services. AVs, relying solely on their sensors, operate as independent entities. They have their own database, modeling their behaviors and making decisions autonomously. Modeling the behaviors of other road users enhances interaction, while self-behavior modeling fosters better self-understanding.

The digital world is primarily designed for CVs and CAVs, enabling them to upload and download information. By bridging the physical and digital worlds, a Digital Twin (DT) of each CV or CAV can be created through direct communication. For AVs and HDVs, their DT can be built by utilizing information sensed and shared from CVs or CAVs. Therefore, in the digital world, each entity has access to a vast amount of data and shared knowledge, enabling enhanced cooperative prediction, cooperative motion planning, personalized services, and efficient interactions among vehicles, other road users, and infrastructure. These capabilities are facilitated by many algorithms, from the behavior module and planning module of the shared knowledge. This dissertation aims to realize this proposed system and develop the algorithm for IVs operation to improve the safety, mobility, and environmental sustainability of the whole traffic system.

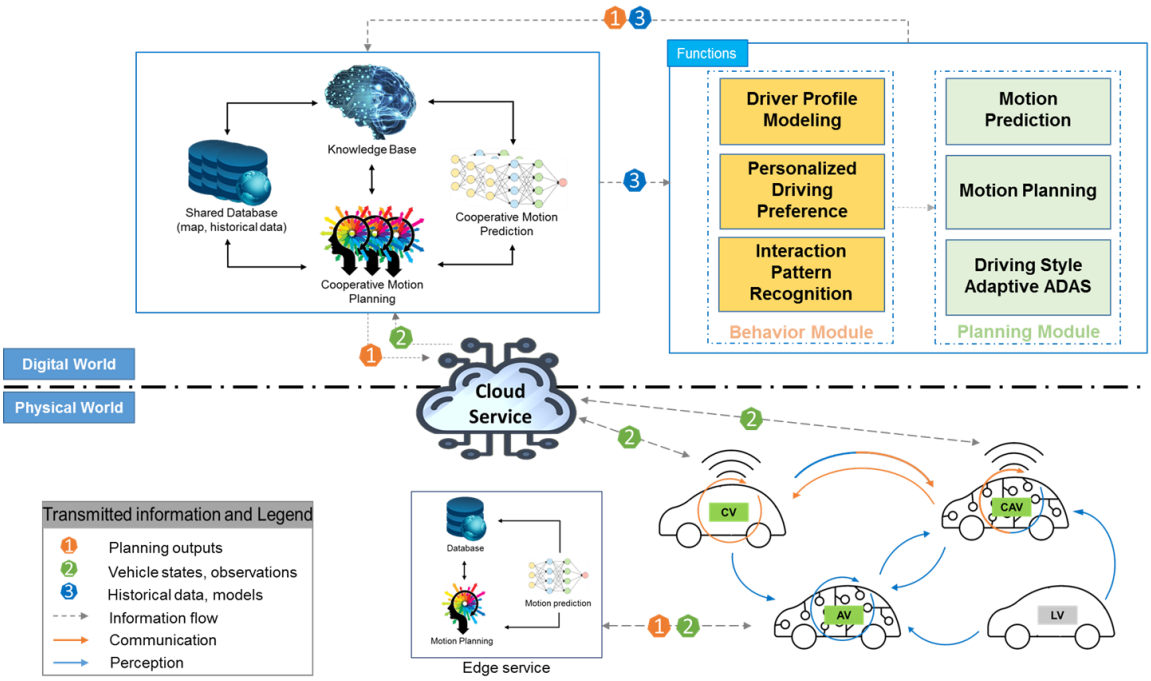


Fig. 3-1 Personalized behavior-aware motion planning framework for IV at traffic system level

3.1 Intelligent Vehicle Operation Pipeline

From a single vehicle perspective, IV is operated in a sequential method shown in Fig. 3-2, including the communication, perception, localization, sensor fusion, planning, control, and actuator components. The IV communication module facilitates reliable, real-time vehicle-to-everything (V2X) wireless communication. Perception sensors equipped on AVs and CAVs serve as the primary data sources for situation awareness of surrounding vehicles and road conditions, feeding this information to the sensor fusion component. Localization functionality of an IV system integrates two distinct hardware components: the global navigation satellite system (GNSS) and the inertial navigation system (INS), along with map matching. The sensor fusion stage aggregates and synthesizes information

gathered from communication, perception, and localization, providing a holistic and in-depth understanding of the surrounding environment.

The planning module is the focus of this dissertation, and it processes all the acquired data to perform the necessary tasks (as introduced in Section 2.2) of path optimization, motion prediction, decision-making, trajectory planning, and the formulation of safe and efficient driving strategies. Also, it is the part where the behavior module (i.e., driver modeling and personalization) is incorporated to develop the optimal and favorable strategies, tailored to individual driver habits and preferences, enhancing the overall user experience and acceptance of the automated driving system.

The final phase of this sequential process involves the control and actuator modules. The control module in an IV system is comprised of both software and hardware elements: a controller component that enables the motion control algorithms of the IV, and the physical actuators within the IV that execute the longitudinal and lateral commands as directed by the controller component.

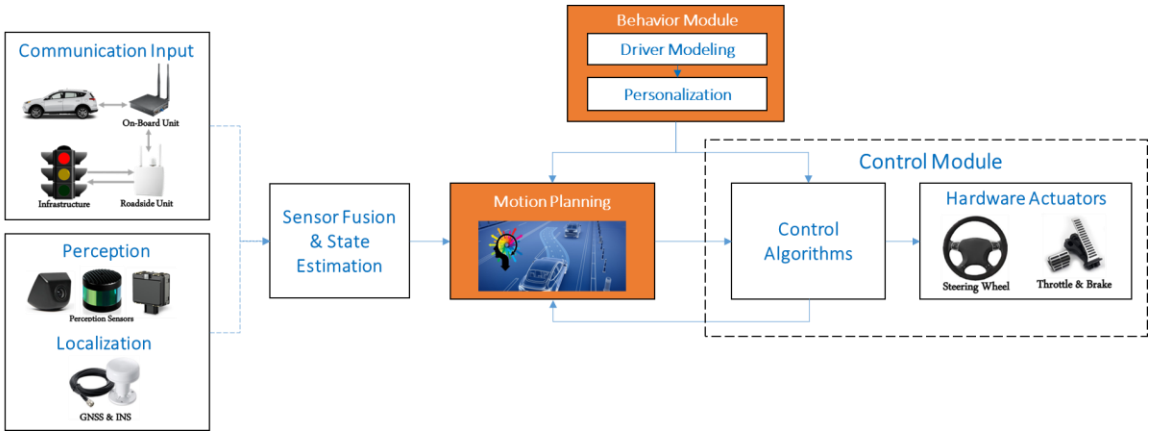


Fig. 3-2 Operation pipeline of intelligent vehicles with the integration of behavior modeling

3.2 Integration of Behavior Modeling

This section elaborates on the design of a behavior-aware motion planning system for IVs in both fully connected traffic and mixed traffic. The proposed system is designed from a decentralized agent-based model perspective for IV, allowing vehicles to act independently. For algorithm development and verification, we establish a database, a local server for real-world algorithms implementation, and a Human-in-the-Loop (HuiL) co-simulation platform. The strategy workflow is depicted in Fig. 3-3, with each vehicle navigating through the modules, represented by solid lines, at every time step.

1) *Driving Behavior Modeling*: This module consists of a behavioral layer to understand the involved driver's behavior. Modeling the driver's behavior is usually an offline process and can facilitate the downstream prediction and planning modules in real time, as indicated by the red solid lines.

2) *Maneuver and Trajectory Prediction Module*: This module uses inputs from the behavior module and sensors for predictions. First, this module judges whether conflicts exist in the future and identifies the vehicle type of competitor (e.g., CAV, CV, AV, or HDV). Then it anticipates the type of maneuver the driver will execute, as well as when and where it will occur. Moreover, it predicts the trajectory required to complete the anticipated maneuver.

3) *Interactive Planning*: This module captures the interaction between the ego vehicle and its surroundings. Based on the prediction results and the understanding of the other driver's behavior, this module generates an optimal and favorable trajectory for the control module.

4) *Driving Style Adaptive Planning for Ego Vehicle*: As the Interactive Planning module deals with the interaction with other road users, this module considers more about the ego vehicles. By learning the driving preference of ego vehicle's driver, it provides suggestions to adjust the planning strategy for improving the user experience. Furthermore, this module allows extending the algorithm of IV to HDV and will be introduced in Section 4.3.3.

5) *Acceleration Control Module*: This module is responsible for ensuring the ego IV runs at the desired speed and tracks the lane. It is colored gray because it is out of the research scope of this dissertation.

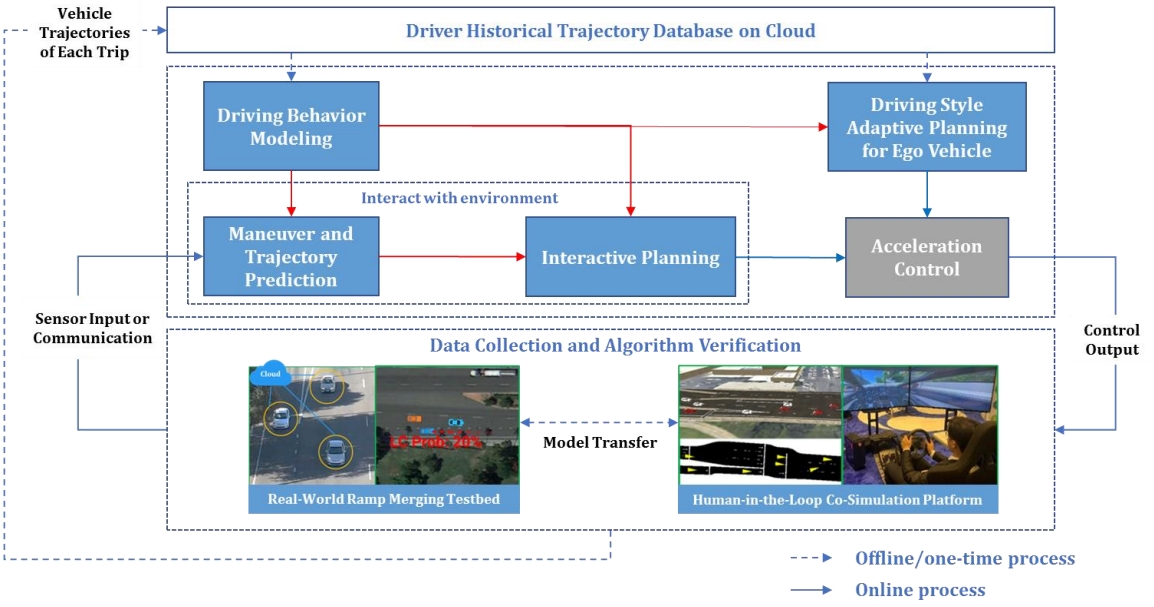


Fig. 3-3 Schema of the proposed personalized behavior-aware motion planning framework for IV at vehicle level

4 Methodologies

4.1 Platform Construction

4.1.1 Vehicle-Edge-Cloud (VEC) Real-World Implementation Platform

4.1.1.1 Introduction and Background

To study personalized driving behavior and develop motion planning algorithms for IVs, this dissertation deploys a Digital Twin (DT) approach in the real world. A DT of an intelligent vehicle is a virtual representation of the vehicle that mirrors its physical counterpart in real time. A driver DT (DDT) is a digital replica of a driver with his or her naturalistic driving data and driving behavior models. DT enables vehicle-to-everything (V2X) communication, driver behavior analysis, personalization, traffic and infrastructure planning, etc. The implementation of DT uses real-world data, machine learning, and software analytics to simulate, predict, and visualize the vehicle's performance, needs, and challenges.

The realization of DT is achieved by a Vehicle-Edge-Cloud (VEC) platform, which is introduced in Fig. 4-1, including key components, hardware for the real-world implementation, and the information flow between them. Under this architecture, vehicles are considered service consumers who store their personalized dataset on the cloud server and share the real-time perception information with the edge server, from which vehicles receive driving assistance and support services to facilitate automation. The edge server creates the vehicle digital twin and serves as the bridge between the cloud server and vehicles. DDT is created on a cloud server by offline modeling algorithms.

Based on real-world data, DDT in the virtual world provides both online and offline micro-services, e.g., interactive prediction, driving style analysis, etc. In addition, to capture the driving preference variation, the evolving driver model will be updated in a certain period (e.g., every five new trips) by consuming the driving data from the real-world vehicle. Supported by the edge server, VEC allows the planning and control of the connected vehicles while ensuring real-time computation. However, only using an edge server is sometimes insufficient to fulfill the requirements of personalized behavior study, such as data storage, modeling, learning, simulation, and prediction. Therefore, behavior study in this dissertation is performed on a cloud-edge architecture, leveraging cloud computing and personalized profiling, enabling both real-time and bulk-batch ingestion, processing, and analytics of personal data. Based on real-world data, VEC in the virtual world provides both online and offline micro-services, e.g., interactive prediction, driving style analysis, etc. In addition, to capture the driving preference variation, the evolving driver model will be updated in a certain period (e.g., every five new trips) by consuming the driving data from the real-world vehicle.

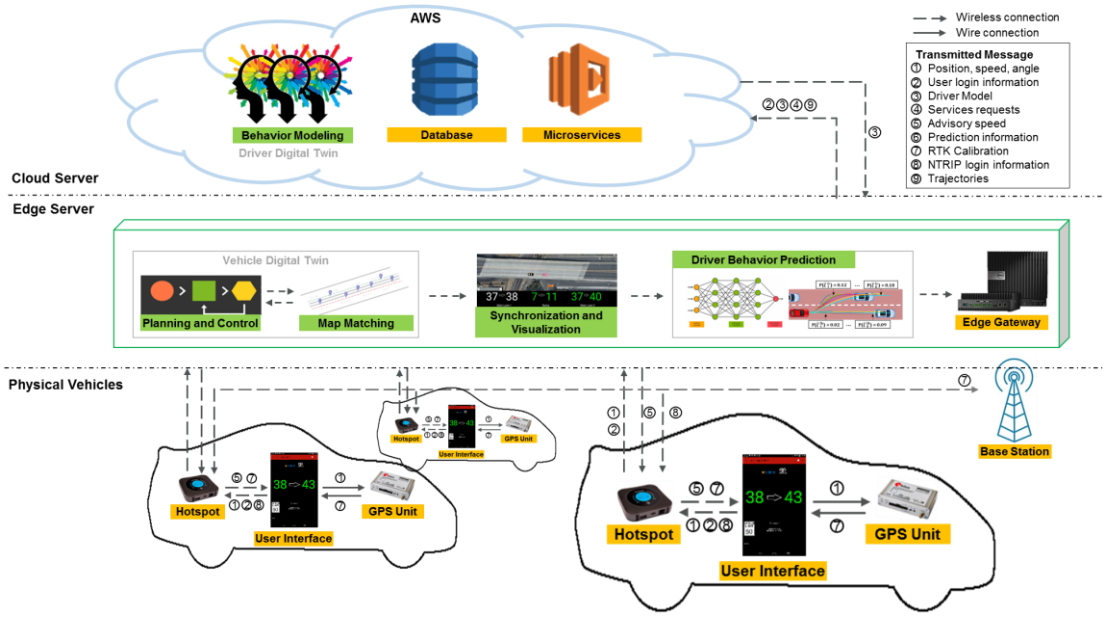


Fig. 4-1 The vehicle-edge-cloud platform architecture validated in the field experiment

4.1.1.2 System Setup

To be specific, the first step of building the vehicle digital twin on the edge server is to know the position of each vehicle in the network. After building the connection between edge and vehicles, the perception information and GPS coordinates of each vehicle are uploaded to the edge server, where the map-matching algorithm updates the vehicle's positions in real time. Furthermore, algorithms for predictions, planning, and control can be provided upon request for different purposes. With the connection to the cloud server, the edge server receives service requests from vehicles and passes on those requests and the unique login information to the cloud. In response to the request, the cloud can feed back edge server with personalized driving behavior models of specific drivers. Due to the high communication latency between the cloud and vehicles, it is hard to implement real-time services provided directly from the cloud to customers. To address this problem, an

Edge Gateway is adopted to handle the data exchange and provide services to vehicles in real time.

The concern of privacy is also addressed, as only unidentifiable information is transmitted. The cloud server is responsible for driver model training, storage of personalized data and models, and microservices support (e.g., energy consumption analysis). Therefore, we take advantage of the strong computational power, high-speed data processing, and secure data storing features of AWS. Historical personal driving data are archived in the personal folder and can only be accessed by the personal login information. Service consumers can also request post-processed driving behavior reports, (e.g., energy consumption analysis, to better understand or improve their driving skills.

Hardware for the real-world implementation is shown in Fig. 4-2. All three vehicles are 2012 Corolla LE models, with 1.8-liter internal combustion engines. Each vehicle is equipped with a Wi-Fi hotspot (Netgear MiFi) to establish a wireless connection with edge server, a GPS unit to collect accurate position information of vehicles, and a portable human-machine interface (HMI) device using Galaxy A7 tablet. The U-blox C102-F9R is adopted as the GPS unit, which is a multi-band GNSS with Real-Time Kinematic positioning (RTK) function and sensor fusion technologies. It can achieve a 2-dimensional (horizontal) accuracy of 46.7 cm and 4.0 cm with the “RTK Fixed” mode [87]. The GPS unit and the HMI device are connected (wired) with serial communications. The Wi-Fi hotspot handles the wireless communications between the tablet and other infrastructures such as the base station and edge server. Depending on the request of the driver, the HMI device receives and displays the speed guidance or lane-change prediction information

from the edge server. Moreover, to enable the RTK function in the U-blox and get high-accuracy GPS measurements, the HMI device shares the calibration correction message received from the base station to its paired GPS unit.

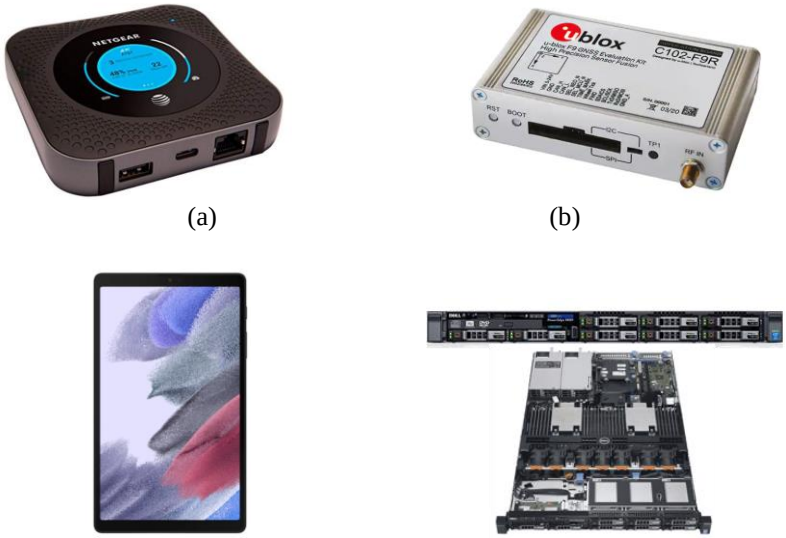


Fig. 4-2 Hardware in VEC platform: (a) NETGEAR hotspot (b) U-blox C94-M8P-2 GNSS unit, (c) Google Nexus 7, and (d) Dell R630 server

The edge gateway is running on a customized edge server installed at CE-CERT, University of California, Riverside, which is 1.2 km away from the testing site. A Dell R630 server, with two Xeon 2.4GHz (6-core) CPUs, 64GB RAM, 1TB solid-state drive, and 14 TB hard disk drive, is adopted as the edge server in this implementation, as shown in Fig. 4-2 (d).

Moreover, the communication delay and computation time are considered for real-world implementation. The back-and-forth communication delays of cloud-edge communication

(model download), the edge-vehicle communication (speed guidance and online prediction), and the computation time are measured and shown, respectively, in Table I.

TABLE I STATISTIC RESULT OF COMMUNICATION DELAY AND COMPUTATION TIME

	Average	75th Percentile	99th Percentile	Maximum
Cloud-Edge	800 ms	915 ms	1221 ms	1261 ms
Edge-Vehicle	80 ms	88 ms	247 ms	3861 ms
Computation	28 ms	14 ms	36 ms	71 ms

4.1.1.3 Hidden Markov Model Map Matching

Map matching is an important component in a field implementation to reduce the effect of noisy GPS measurements. Hidden Markov Model (HMM) exploits the road connectivity information and time-sequence feasibility to solve the problem. As shown in Algorithm 1, we briefly introduce the HMM map-matching algorithm, which was proposed by Newson and Krumm [92]. For the HMM map matching process, we first predefine a road network with a set of geographical coordinate pairs (i.e., latitude and longitude) of the road segments based on the real-world situation. The input is the GPS measurement points from the current time step and the last time step. The goal of map matching is to find the best-fitted road segment from all candidate road segments. The candidate road segments are selected by the great-circle distance between the measurement points and projection points on the road segments. Among candidates, we find the one which maximizes the product of the measurement probability and transition probability. Then we output the projection point of the current GPS measurements on the best-fitted road segment as the matching point and the distance between them.

Algorithm 1: HMM Map Matching Algorithm

Input: 1. Current GPS point (z_t). 2. Road segments on map (r_i)

Output: 1. Matching point (r_{best}). 2. Distance to road

1: if (t = 0) Initialize z_{t-1}

2: Find candidate road segments (r_i) based on z_t

3: Calculate the projection $x_{t,i}$ of z_t on each r_i

4: Calculate the probability of the measurement $(z_t|r_i)$ by

$$(z_t|r_i) = \frac{1}{\sqrt{2\pi}\sigma_z} exp\left(-0.5\left(\frac{\left\|z_t - x_{t,i}\right\|_{great\ circle}}{\sigma_z}\right)^2\right)$$

5: Calculate the transition probability $p(d_t)$ by

$$p(d_t) = \frac{1}{\beta} exp\left(-\frac{\left|\left\|z_t - z_{t-1}\right\|_{great\ circle} - \left\|x_{t,i} - x_{t-1,j}\right\|_{route}\right|}{\beta}\right)$$

6: Find the best-fitted road segment by

$$r_{best} = \underset{i}{argmax}\left(p(z_t|r_i) * p(d_t)\right)$$

7: Find the projection $x_{t,\text{best}}$ of z_t on r_{best}

8: Calculate the distance between z_t and $x_{t,\text{best}}$ by

$$\left\|z_t - x_{t,\text{best}}\right\|_{great\ circle}$$

9: Update state for next coming measurement $z_{t-1} = z_t$

Return r_{best} and $\left\|z_t - x_{t,\text{best}}\right\|_{great\ circle}$

The test track consists of a ramp and a mainline, where the mainline spans from the intersection of Columbia Avenue and Chicago Avenue to the intersection of Iowa Avenue in Riverside, California. The mainline is on an overpass while the ramp is under the overpass, which increases the difficulty of ramp merging because the vision of the driver is blocked by the construction. The total length of the track is 780 meters, with a merging zone of 89 meters long, which is encoded as road segments in GPS points for the usage of HMM map matching.

4.1.2 Human-in-the-Loop Co-Simulation Platform

4.1.2.1 Introduction

In this dissertation, a human-in-the-loop co-simulation platform[93], is utilized, where the real-world test track is programmed in the Unity game engine as a digital replica of the physical testbed. The simulation incorporates a mixed traffic flow generated by SUMO, while human input is integrated through the use of a Logitech driving set, as depicted in Fig. 4-3. This setup enables drivers to engage in immersive human-in-the-loop simulations within a realistic traffic environment, allowing for the evaluation of various scenarios involving different levels of connected and automated vehicle (CAV) penetration rates and congestion levels. By leveraging this simulation framework, the dissertation aims to facilitate comprehensive and accurate modeling of driving behavior, ensuring the development of robust and effective algorithms for real-world deployment.

Fig. 4-3 Human-in-the-Loop Unity-SUMO co-simulation based on a real-world ramp merging area in Riverside, CA

4.1.2.2 Specifications

To create a realistic mixed traffic flow for the human-in-the-loop (HuiL) simulation, we define three general vehicle types participating in the simulation, which are legacy vehicle (background vehicle), IV, and user control vehicle. When SUMO is responsible for legacy vehicles, IVs and user control vehicles are using the vehicle model in Unity.

For longitudinal control, the SUMO controlled legacy vehicles adopts the Krauss car-following model and IDM [94]. To avoid homogeneous driving behaviors, the imperfection parameter is set to be 0.5. In addition, the speed deviation is set to be 10%, resulting in a speed distribution where 95% of the vehicles would travel at a speed ranging from 90% to 110% of the legal speed limit. Besides the car-following model embedded in SUMO, Unity provides API allowing the customized algorithm for IV control.

For lateral behavior model, the default collision-free lane change model in SUMO developed by Erdmann [95] controls the lateral maneuver of legacy vehicles, and the default parameters for passenger vehicles are adopted in our simulation. In Unity, lane change maneuvers are governed by Unity's built-in Bezier curve generator. In the simulation, each lane has its reference path consisting of waypoints for every CAV to track. If gaps are determined to be safe, then CAVs can start their lane changes which can be triggered by moving the future waypoints from one lane to another. As shown in Fig. 4-4, the CAV just passes Waypoint 23 and is ready to perform a lane change. Based on its current speed, it first selects the finishing waypoint (Waypoint 25) of lane change on its left (target) lane. Next, the Bezier Curve generator creates Waypoint 24 between Waypoint 23 and Waypoint 25, and connects those waypoints with a smoothed path.

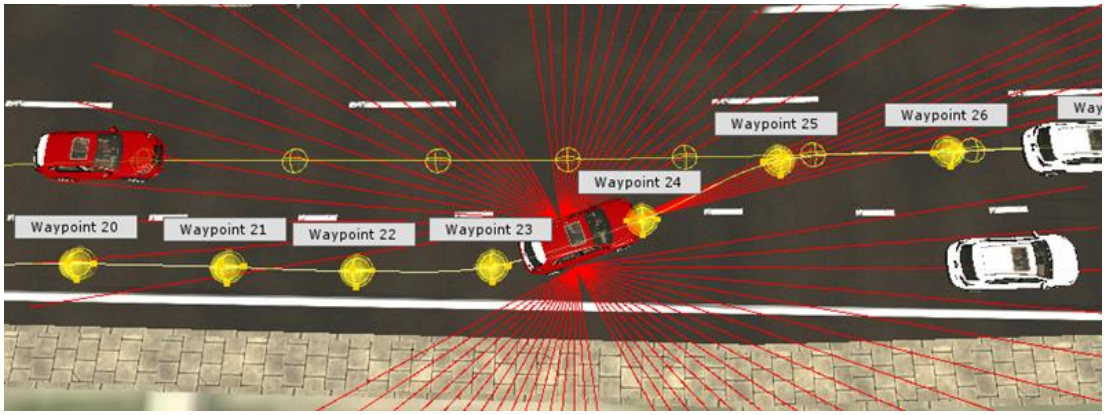


Fig. 4-4 Bezier curve generator for CAV lane change maneuver.

Moreover, a customized perception system is equipped on IV in Unity, as shown in Fig. 4-4. IVs are color-coded in red, while legacy vehicles are in white. The red rays spread from the IV indicate the detection range of its onboard radar system, where the long-range radar in the front has a 150m detection range and 18-degree field of view, while short-range radars on the side have a 25m detection range and 70-degree field of view. The

characteristics and performance parameters of this onboard radar system are selected based on off-the-shelf radar sensors [96].

Finally, a two-way communication via UDP Socket connects and synchronizes these two simulation platforms in real-time, allowing SUMO to control legacy vehicles while Unity controls IVs with the customized algorithm. After each trip, the driving data containing information of each vehicle is uploaded to AWS for training and storage.

4.2 Personalized Driving Behavior Modeling

This section explores the intricate complexities of individual driving profile and emphasizes the importance of understanding and predicting the decisions drivers make based on their unique driving preference. The tripartite structure of this section comprises an exploration into driver profile, a detailed analysis of interaction pattern, and a deep dive into driving preference. Each subsection takes a closer look at the integral components of personalized driving behavior and offers insights into how these unique patterns can be accurately modeled and incorporated into intelligent vehicular systems to optimize both safety and efficiency.

4.2.1 Driver Profile Modeling Based on Personality, Driving Style, and Mood States

4.2.1.1 Introduction and Background

Recent developments in automated driving technologies will result in road interactions between automated vehicles (AVs) and human-driven vehicles in the foreseeable future. With these advancements, the role of the driver is likely to change. This presents unique challenges to driver safety and driver state assessment. To understand the critical safety issues that drivers may face in the future, it is essential to understand driver profiles.

Compared to the existing literature on prediction and behavior modeling, our study has made several contributions:

- A comprehensive framework to evaluate driving styles and their corresponding mood states was developed. The driving simulator provided a controlled environment to guarantee all participants experience the same scenarios and well-defined events.
- A longitudinal user study was designed, and data collection was conducted to integrate the driving style, personality traits, and mood state of each participant into a single dataset.
- Alongside a prediction model for driving style that uses mood states and personality traits, an inference model for personality types (obtained by clustering) given mood states and driving style was developed and assessed for accuracy.

The findings from this study are applicable in two major implementations. 1) Driving style prediction- A risky driving style can be predicted to adopt a new Advanced Driver Assistant System (ADAS) strategy, given the personality traits and mood states of the driver, and the mood states can be determined by smart devices and/or a driver-monitoring camera. 2) Personality type inference can help personalize the product setup when drivers use other mobility anywhere (i.e., Mobility-as-a-Service), by observing users' driving style and mood states. In this study, such inference is used interchangeably with prediction.

4.2.1.2 Problem Formulation

Tenets of personality traits, mood states, and driving style were assumed as follows: 1) Personality traits are enduring attributions of each participant and consistent during the experiment period. 2) Mood states are affected by participants' experience near the experiment day, which can be different per experiment session. 3) While each participant's baseline driving style (e.g., aggressive) is enduring, it is influenced by mood states.

In this section, the experimental design, data collection, and metrics used for the analysis are discussed.

4.2.1.2.1 Experiment Design

A total of 28 individuals between the ages of 21 and 40 (Mean = 27.53 years, SD = 5.06 years) were recruited in California's San Francisco Bay Area by way of online advertisements as well as at physical locations. All participants were male, fluent in English, and had a valid U.S. driver's license. Prior to the user study, each participant took a personality traits assessment. During it, they would visit the test site four times altogether with a one-week interval between each visit. In every visit, participants experienced two driving simulator sessions: an urban city and a highway driving scenario. A mood check was required afterwards. For two out of four visits, each participant was asked to write down their happy and angry experiences and played an audio track. This was done to facilitate mood manipulation as music, imagination, and recall have been found to induce mood [97]. A sum of 224 driving session runs was carried out.

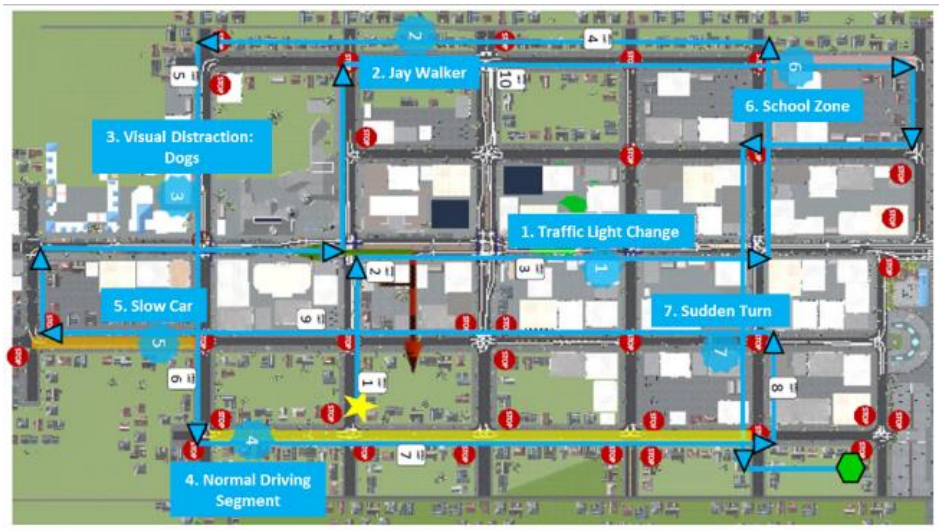
4.2.1.2.2 Integrated Dataset Construction

A. Personality Traits. To assess personality traits, the NEO Personality Inventory-3 (NEO-PI-3) questionnaire developed by McCrae et al. [98] was employed. This test is modeled upon the Five-Factor Model [99] as the basis of human personality. The five traits (i.e., Big-Five Scores) are Neuroticism, Extroversion, Openness, Agreeableness, and Conscientiousness. The evaluation of each personality trait is based on six sub-scores, evaluated by eight survey questions. The raw scores of the five factors and their sub-scores were calculated based on the survey response, and then for each trait, standardized T scores [100] were used. The T score represented the standardized values for each personality trait, where a score of 50 represents the mean and a difference of 10 from the mean is the difference of one standard deviation.

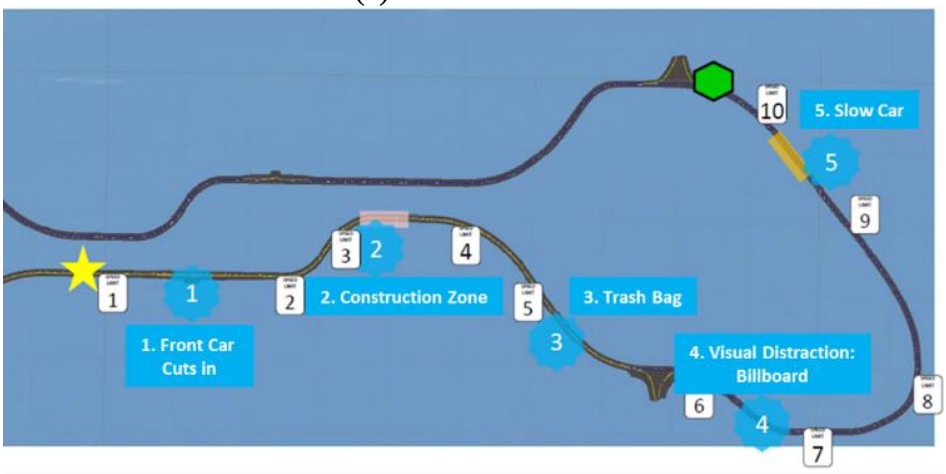
B. Profile of Mood States. The mood profile of a driver was assessed using the Profile of Mood States 2nd Edition-Adult Short (POMS 2-A Short) survey, developed by Terry et al. [101]. This 24-item questionnaire was adopted because of its ability to capture transient and fluctuating feelings. The responses of the assessment produced eight factors, including scores for six mood clusters: Anger-Hostility (Anger), Confusion-Bewilderment (Confusion), Depression-Dejection (Depression), Fatigue-Inertia (Fatigue), Tension-Anxiety (Tension), and Vigor-Activity (Vigor). Two general scores were also generated: Total Mood Disturbance (TMD) and Friendliness.

C. Driving Simulation. Unlike mood states and personality traits, driving style is evaluated objectively from participants' driving behavior. To understand driving style, the driving trajectories for each participant were observed using an in-house driving simulator

built with AirSim [102], a plug-in for Unreal Engine 4. The participants controlled the vehicle using a Logitech G29 steering wheel and pedals.



(a) Urban scenario



(b) Highway scenario

Fig. 4-5 Events design for driving style evaluation

The driving simulations presented urban city and highway scenarios to participants, as shown in Fig. 4-5. In each scenario, drivers experienced five critical events which included sudden danger (e.g., sudden cut-in behavior), speed limit signs, a slow preceding vehicle, visual distractions, and normal driving. Driving trajectories included vehicle coordinates, principal axes (yaw, pitch, rotation in degrees), speed (miles per hour), throttle and brake

(0 - no throttle/brake to 1 - full throttle/brake), steering angle (0 to 1), distance from lane center (in meters), and the distance from surrounding vehicles (m). Combining driver reactions under different events, each session was classified into a certain driving style by the proposed fuzzy-logic inference system.

D. Data Analysis. The driving trajectories in each session were classified into different driving styles, and the mood states and personality traits were processed. Principal Component Analysis (PCA) was applied for feature selection of the mood data. In assessing personality traits, Hierarchical Clustering Analysis (HCA) helped to cluster drivers with similar personalities. This was done because initial analysis found (1) regression for five-dimensional traits on a small-sample dataset was impractical and (2) a large number of combinations of five continuous traits were infeasible for classification. Furthermore, data cleaning was executed to filter out unrealistic sessions (e.g., driving on the sidewalk in the city scenario) and 201 out of 224 data points remained in the dataset.

4.2.1.3 Methodology

In this section, the algorithms for profiling the mood states, personality traits, and driving styles of participants are described. As shown in Fig. 4-6, this study consists of a data collection phase and a modeling phase. In the data collection phase, each participant followed the experimental procedures as their mood states, driving trajectory, and personality traits were collected. As the one important part of driver profile modeling, the correlation between mood states and personality traits was investigated using their scores.

In the modeling phase, training and test datasets were split. For the assessment of mood states, three principal components from mood states explained 93% of mood states, and

based on the contribution to three principal components, five out of eight significant features (i.e., Tension, Vigor, Fatigue, Friendliness, and TMD) were selected. Four driving styles were determined by the fuzzy logic inference system based on driving trajectories and three personality types were clustered by HCA. Eventually, a prediction model was trained and validated by random forest, enabling the prediction of (1) driving style with mood states and personality traits and (2) personality types with mood states and driving style.

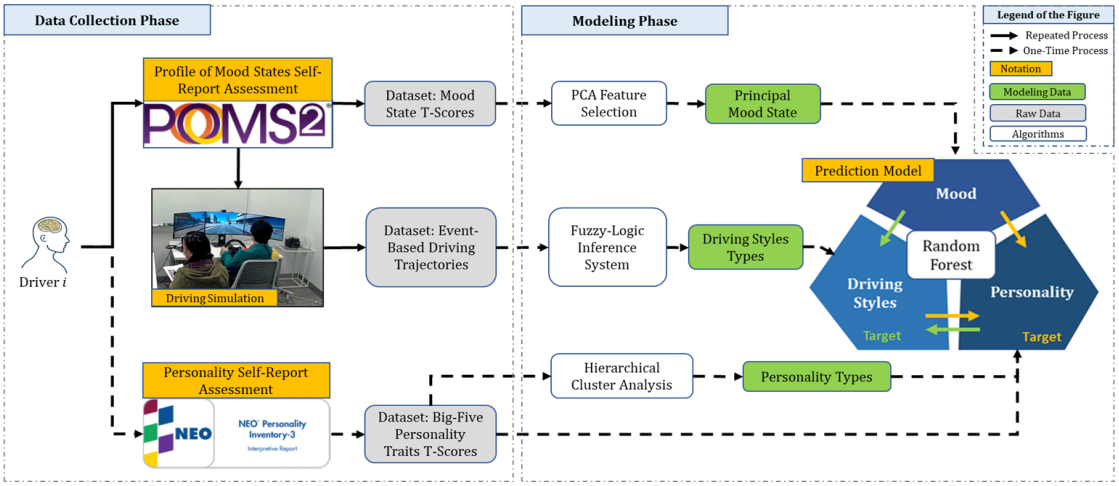


Fig. 4-6 System workflow of the driver profile modeling (data collection and modeling)

4.2.1.3.1 Enhanced fuzzy logic inference system for driving style recognition

In this study, four driving styles were defined based on definitions given by [24], and they were aggressive, anxious, keen, and sedate. Economical type was excluded because the behavior based on fuel consumption could not be replicated in a simulator study, also participants did not feel time efficiency concerns. The driving styles are not mutually exclusive, but there were predominant styles. The recorded 201 sessions were classified into one of the four styles. To utilize prior knowledge of driving trajectory, the fuzzy logic

inference system was adopted to classify driving styles by interpreting the fuzzy linguistic terms given by the definitions. To ensure separation between driving styles, the weights used in the fuzzy logic inference systems were optimized.

Given the driving trajectories collected in the simulator, the fuzzy logic inference system estimated the probability of how each trajectory could be classified into a predefined driving style. The classification was done based on the highest probability. To be specific, the fuzzy logic inference system evaluated drivers' reactions to well-defined events and final probability was calculated by the weighted sum of each reaction. For example, an average speed of 110 mph in a session would be labeled as Very High, and the probability that the driving style is typified as aggressive may increase, and at the same time that the probability of it being anxious may decrease.

TABLE II EXAMPLE FUZZY-RULES FOR HIGHWAY SCENARIO

		Aggressive	Anxious	Keen	Sedate
Speed	Low	NL	VL	HL	L
	Medium	HL	HL	VL	VL
	High	L	NL	L	NL
	Very High	VL	NL	NL	NL
Brake	Light	VL	HL	HL	L
	Medium	L	NL	VL	HL
	High	HL	VL	L	NL

*NL - Not Likely, HL - Hardy Likely, L - Likely, and VL - Very Likely.

Considering the difference in driving trajectories between city and highway scenarios, two corresponding sets of fuzzy rules were developed for each scenario type to analyze the reactions in events, including normal driving (cruising without surrounding vehicle), car following [103], stop sign approaching and departure [24], and lane change [104]. In the city scenario, intersections and normal driving accounted for the majority of the scene. To

evaluate how the participants performed on city roads, four key features were selected: 1) average speed near speed limit signs, 2) minimum speed at stop signs, 3) maximum acceleration after stop, and 4) maximum deceleration when approaching stop signs [105]. In the highway scenario, driving style was analyzed by its interaction with surrounding vehicles and normal driving, and hence four features selected from different events were evaluated: 1) average speed near speed limit signs, 2) maximum brake force when another vehicle cuts in, 3) minimum time headway to the preceding vehicle [106], and 4) lane change rate (i.e., lane change occurrence per mile) [107]. Based on predefined fuzzy rules, the inference system quantified linguistic probability (i.e., from not likely to very likely) into probability values. Due to article length restrictions, only 7 of 30 example fuzzy rules were shown in Table II.

The probability of each driving style could be expressed as Equation (1), where a weight factor $w_{ds,f}$ was introduced to define how much a feature (f) contributes to a particular driving style [24]:

$$p(ds) = \sum_{f \in \text{ features}} w_{ds,f} \cdot p(ds \mid f) \quad (1)$$

where $ds \in DS = \{Aggressive, Anxious, Keen, Sedate\}$, and $\sum w_{ds,f} = 1$.

To avoid ambiguities in classification between similar driving styles (e.g., aggressive with keen, anxious with sedate), Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [108] was adopted to optimize the weights $w_{ds,f}$. As presented in Equation 2, two objective functions were maximized by tuning the weights. F_1 is the sum of the probability difference between each pair of driving styles, and F_2 is used to find the probability of the most

probable driving style. This optimization process improved classification certainty by maximizing both F_1 and F_2.

$$\begin{cases} F_1 = \sum_{i=1}^{3}\sum_{j=i+1}^{4} \left\| P(DS_i) - P\left(DS_j\right) \right\|_2 \\ F_2 = \underset{ds \in \mathrm{DS}}{\operatorname{argmax}} \left(\sum_{k=1}^{N} p_k(ds)/N \right) \\ F(w) = \text{maximize } (F_1(w), F_2(w)) \\ \text{s.t. } 0 \le w_{ds,f} \le 1 \end{cases} \tag{2}$$

where $P(DS_i)$ is the combination of probabilities of i -th driving style in each session for all participants, $P(DS_i) = \{ P_1(DS_i), \ldots, P_n(DS_i) \}$, N is the number of sessions to be evaluated, and $p_k(ds)$ is the probability of ds at k-th session.

4.2.1.3.2 Prediction based on random forest

For this study, the prediction was formulated as a classification problem with the characteristics of the dataset taken into consideration; Random Forest[109] was used as the classifier as it could process inputs with categorical variables where input data was a synthesis of categorical variables (i.e., types) and continuous variables (i.e., score values). Random forest can, moreover, reduce over-fitting in a small-sample dataset with Bootstrap Aggregating (Bagging) [110]. Also, because the participants were all young male drivers, the dataset was unbalanced with an unequal distribution of mood states and driving styles. By weighing each class, random forest can account for unbalanced datasets effectively. Additionally, the results from classification are voted by multiple decision trees, thereby improving their robustness.

As shown in the prediction model in Fig. 4-6, when the prediction target was driving style, the inputs were personality traits, personality types (obtained from the HCA), and

mood states. When predicting personality types, the inputs were driving styles and mood states. To improve prediction accuracy, grid search (exhaustive search) with 5-fold cross-validation was used to tune the hyper-parameter of the random forest model. Specifically, three major parameters were tuned: the number of decision trees (n_{tree}), maximum depth of the tree (d_{max}), and the number of features to randomly investigate (n_f). As a result, 1) for driving styles prediction, n_{tree} was 100, d_{max} was 50, and n_f was 3, and 2) for personality types of prediction, n_{tree} was 42, d_{max} was 70, and n_f was 3.

4.2.1.4 Result

4.2.1.4.1 Correlation analysis for personality traits and mood state

Correlations between personality traits and mood states are represented in a heat map created from the correlation matrix. As shown in Fig. 4-7, Neuroticism has a positive correlation (correlation coefficient > 0.3) with Tension, Fatigue, and Total Mood Disturbance (TMD). It, however, also has a strong negative correlation with Vigor (r=-0.49, p<.001) and Friendliness (r=-0.41, p<.001). Extraversion has a positive correlation with Depression, Vigor, and Confusion. Compared with other traits, Conscientiousness is associated with all mood states (|coefficients|> 0.3), plus it is positively associated with Vigor (r=0.3, p<.001) and Friendliness (r=0.35, p<.001). It also has a strong negative correlation with Confusion (r=-0.54, p<.001) and TMD (r=-0.52, p<.001). Not to mention, weak correlations (correlation coefficient < 0.3) are detected for Openness and Agreeableness with mood states. Openness has a weak positive correlation with Vigor and Friendliness but displays a weak negative correlation with Tension, Anger, Fatigue, and TMD. Agreeableness has a weak negative correlation with all mood states.

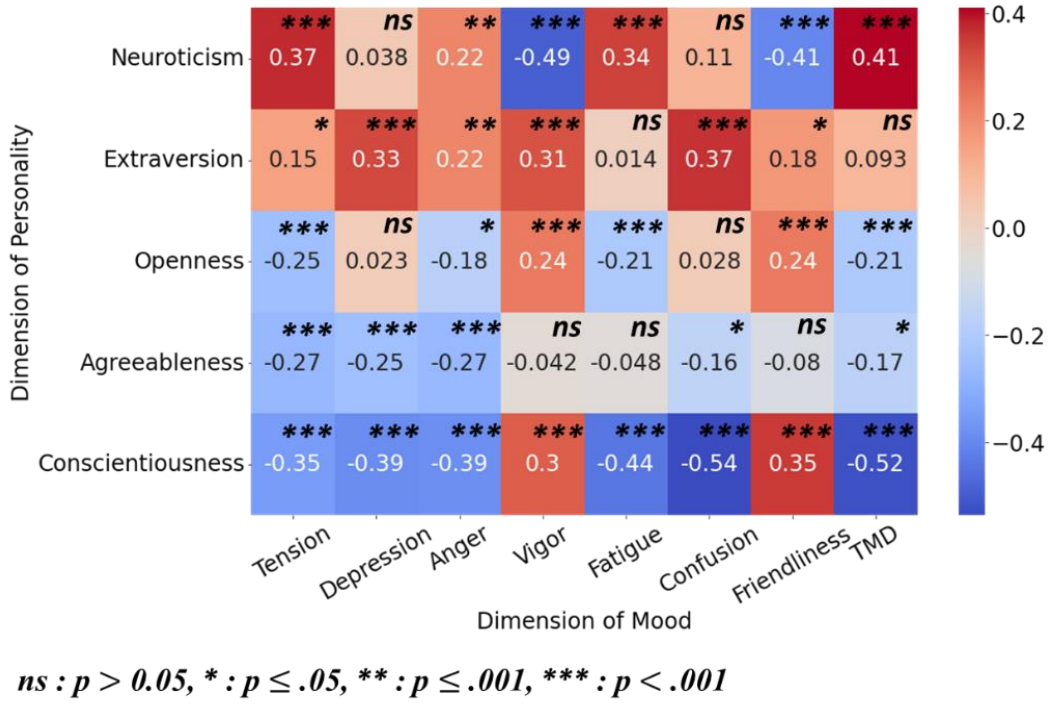


Fig. 4-7 Correlation matrix between mood states and personality traits

4.2.1.4.2 Personality types clustering

Besides the 28 participants, a total of 92 responses to the NEO-PI-3 personality survey were collected online. After applying HCA on personality traits, a dendrogram that represented the Euclidean distance between each data point in a tree-based diagrammatic representation was obtained. This dendrogram suggested a three-cluster result, as shown in Fig. 4-8, where three personality types are colored in orange, green, and red. Notably, not all 92 evaluated participants are shown in Fig. 4-8. The participant count was, respectively, 34, 31, and 27 for the Type 1, Type 2, and Type 3 personality. The average Big-Five personality traits for the three types of personality are shown in Fig. 4-9. Compared to the other two, Type 1 has the lowest scores for Agreeableness and Openness but its score for Neuroticism is high. Type 2 personalities have the lowest Neuroticism scores and high

Conscientiousness, Extraversion, and Openness. The scores on Extraversion and Conscientiousness are low for Type 3 personalities, who also have high Agreeableness and Openness.

The averages of each mood state for the three personality types are presented in Fig. 4-10. The difference between mood states for Type 1 is smaller than it is for Type 2 and Type 3. Type 2 personalities have the highest scores for Vigor and Friendliness, and the lowest scores for Tension, Depression, Anger, Fatigue, Confusion, and TMD. Type 3 personalities have the lowest scores for Vigor and Friendliness.

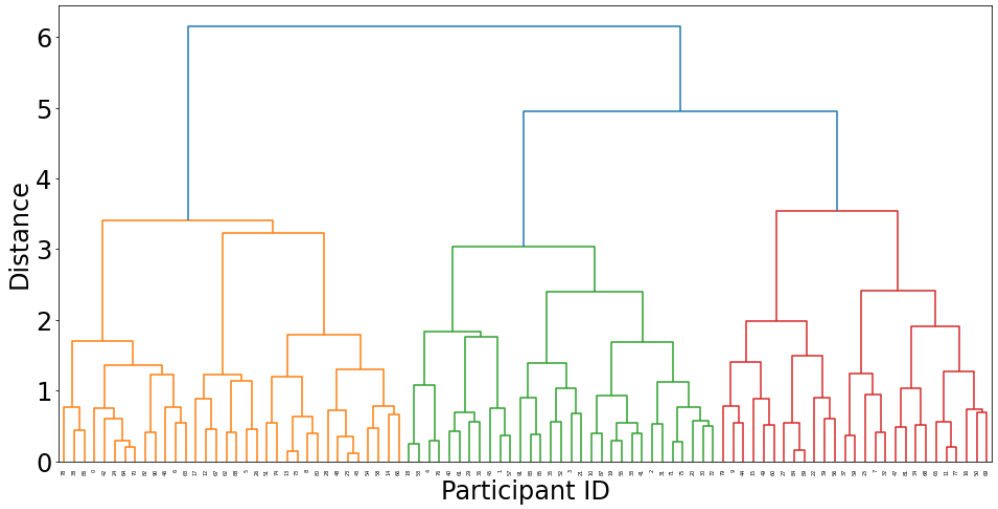


Fig. 4-8 Personality type clustering dendrogram using HCA

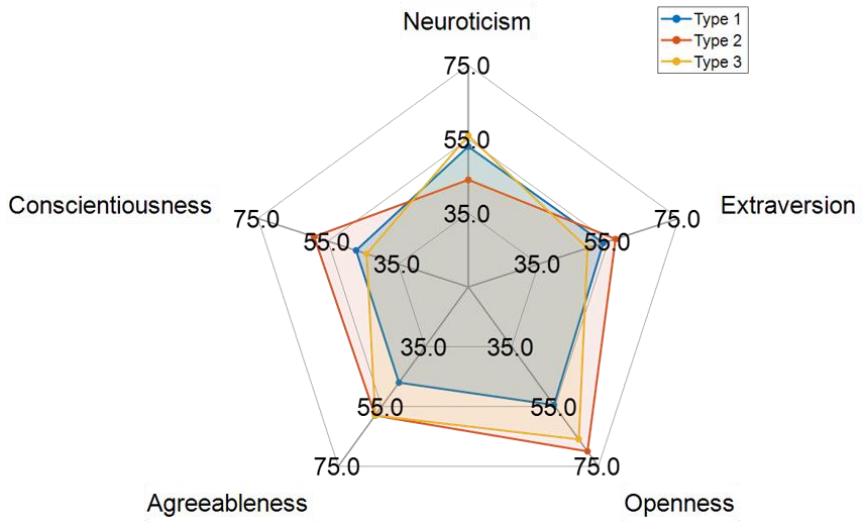


Fig. 4-9 Average scores of three personality types

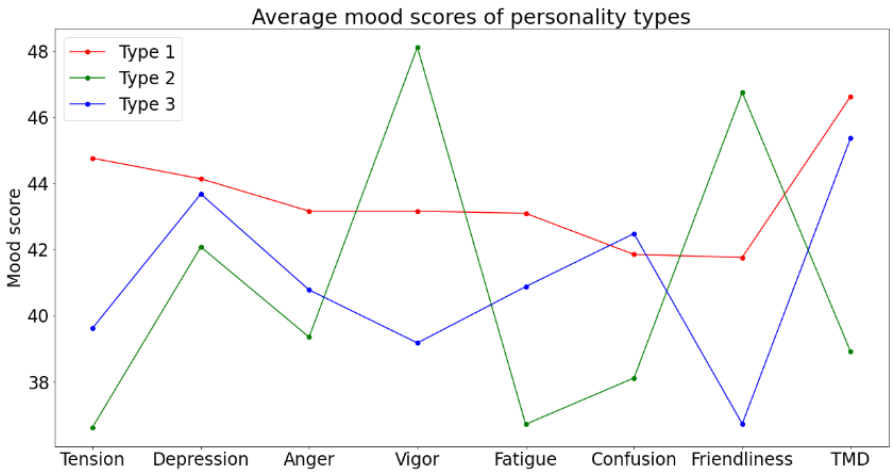


Fig. 4-10 Average mood states of three personality types

4.2.1.4.3 Driving style recognition

The performance of the proposed fuzzy-logic inference system is presented in Fig. 4-11 and Fig. 4-12, where evaluated features from the different driving styles are compared to illustrate the performance of the fuzzy logic system on driving style separation.

In the city scenario shown in Fig. 4-11, aggressive drivers have a higher maximum acceleration after a stop sign and drive 20 mph faster than the speed limit (25 mph) on

average; they usually do not perform a full stop at a stop sign, driving at a minimum speed of 2.84 mph on average. Anxious drivers tend to drive defensively, 1 mph slower than the speed limit on average, and have a low stop sign departure acceleration of 1.59 m/s^2 on average but brake intensely with a deceleration of 3.96 m/s^2 on average. Keen drivers drive 6.79 mph on average faster than speed limit, perform complete stops, and have lower acceleration and deceleration rates than aggressive drivers. Driving defensively, sedate drivers are similar to anxious drivers but focus more on comfort (with the lowest average deceleration of 2.60 m/s^2) and efficiency, which will be better explained in the highway scenario.

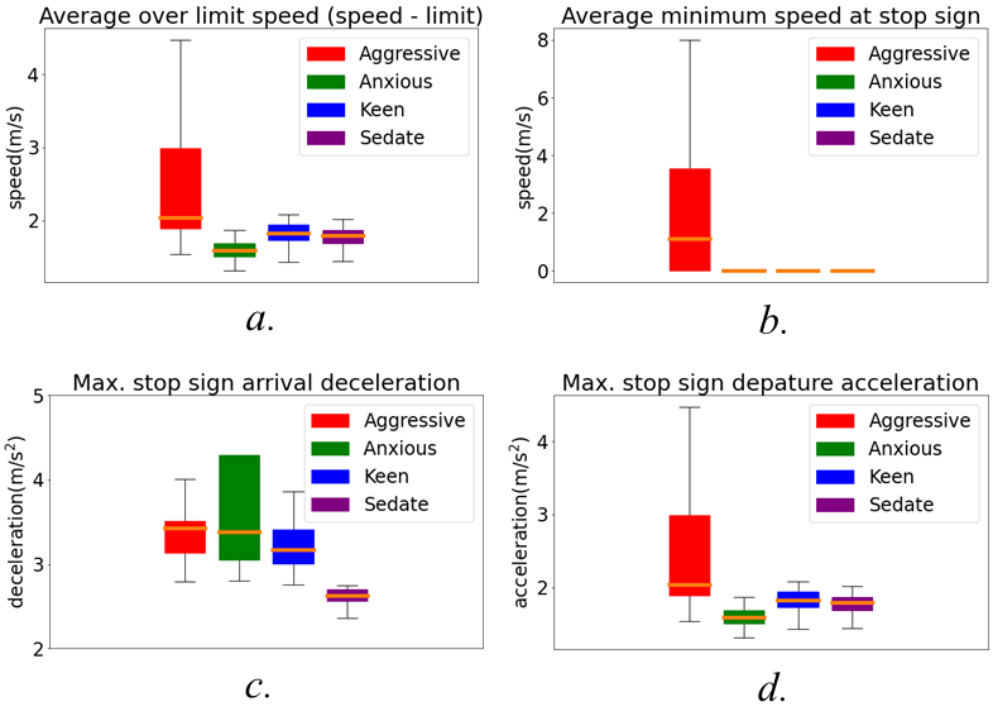


Fig. 4-11 Features in city driving session: (a) Maximum stop sign departure acceleration; (b) Maximum stop sign approaching deceleration; (c) Average speed over the limit (speed - limit); (d) Average minimum speed at a stop sign.

Fig. 4-12 illustrates driver performance in the highway scenario. Aggressive drivers drive 25 mph faster than the speed limit (65 mph) on average, tend to tailgate their

preceding vehicles with a 0.74s average minimum time headway, and change their lane most frequently. Although keen drivers go over the speed limit, they do not threaten their surrounding vehicles (with 0.94s average minimum time headway). Sedate drivers drive at the speed limit (moving 2 mph faster on average), do not perform hard brakes whilst pressing only 32% brake pedal at most, and seldom change lanes (0.21 times per mile on average).

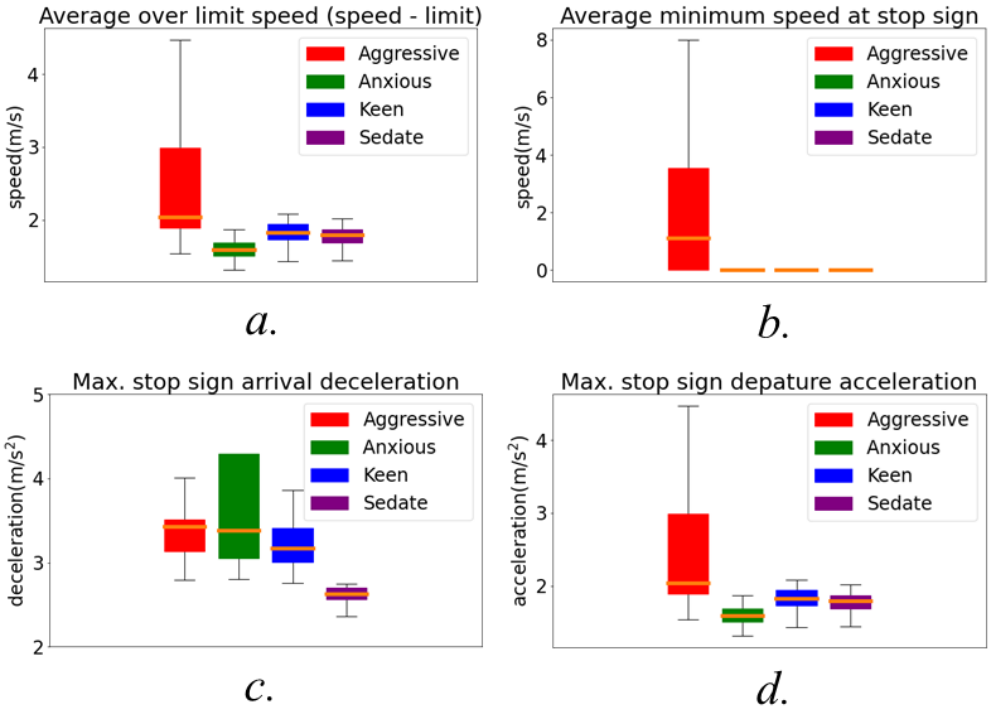


Fig. 4-12 Features in highway driving session: (a) Average speed over the speed limit; (b) Minimum time headway to slow vehicle; (c) Maximum brake force facing cut-in; (d) Lane change ratio of the session.

TABLE III PERSONALITY TYPES-DRIVING STYLES DISTRIBUTION

	Aggressive	Anxious	Keen	Sedate	
P-Type1	12 26.6%	1 2.2%	19 42.2%	13 28.8%	22.5%
P-Type2	14 25.5%	1 1.8%	29 52.7%	11 20.0%	27.4%
P-Type3	36 35.6%	2 1.9%	52 51.5%	11 10.9%	50.3%
	30.9%	2%	49.8%	17.5%	100%

The distribution of different personality type clusters across different recognized driving styles is presented in Table III. Keen driving was observed to be the most frequent style, accounting for 49.8% of all participants while anxious is the least frequent type with only 2%. Among the three personality types, aggressive driving is more frequent for P-Type 3 at 35.6%. Sedate driving is more frequent for P-Type 1 at 28.8%.

TABLE IV PARTICIPANTS' NUMBER OF DRIVING STYLES

# of driving style	$P-$Type1	P-Type2	$P-$Type3	Sum
one	1	1	2	4
two	5	5	9	19
three	1	1	3	5
four	0	0	0	0

Further, during this longitudinal study, many participants showed more than one driving style under different personality types. As shown in Table IV, only 4 of 28 participants insisted on one driving style, 19 drove in two different ways, and 5 showed three driving styles throughout their 8 driving sessions.

4.2.1.4.4 Prediction results

To assess the performance of the predictive models, model accuracy and F1-scores were evaluated. These indices are commonly used for unbalanced datasets. The definitions of these indices are defined as:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \tag{3}$$

$$F_1\text{Score} = \frac{2*\frac{\text{TP}}{\text{TP+FP}}*\frac{\text{TP}}{\text{TP+FN}}}{\frac{\text{TP}}{\text{TP+FP}}+\frac{\text{TP}}{\text{TP+FN}}} \tag{4}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false-negative (predictions). In this study, macro-average scores [111] for these two indices were chosen to evaluate the model performance on the whole dataset.

The random forest predictions, which utilize different data inputs, were compared in Table V. For driving style prediction, using only mood states achieved 0.563 Accuracy and a 0.431 F1-score; a slightly higher result of 0.592 Accuracy and a 0.531 F1-score was obtained with personality traits and types. After combining mood states, personality traits,

TABLE V PREDICTION RESULT EVALUATION

Output	Inputs	F1-score	Accuracy
Driving Style	Mood	0.431	0.563
	Personality	0.531	0.592
	Mood & Personality	0.678	0.696
Personality	Mood	0.131	0.356
	Driving Style	0.597	0.612
	Mood & Driving Style	0.669	0.705

and personality types, the Accuracy and F1-score reached 0.696 and 0.678 respectively. For personality type prediction, although a high correlation was observed as mentioned in Fig. 4-7, Accuracy was merely 0.356 and the F1-score (0.131) was even worse. Using driving styles and their probabilities (obtained from the fuzzy-logic system) led to increased Accuracy (0.612) and F1-score (0.597); in utilizing driving styles and mood states, the greatest result of 0.705 Accuracy and a 0.669 F1-score was acquired.

4.2.1.5 Discussion

The goals of this study were to create an integrated dataset for driver profile modeling, to predict driving style using mood states and personality traits, and to predict personality types using driving style and mood. To the best of our knowledge, prior studies have not combined driving style, mood states, personality traits for the prediction of driving style and personality types.

The correlation between personality traits and mood states found that upon separating mood states into those that were positive (Vigor and Friendliness) and negative (Tension, Depression, Anger, Fatigue, Confusion, and TMD), Neuroticism is positively associated with all negative mood states and negatively associated with all positive mood states. This result is consistent with the previous study[112], which related Neuroticism with negative effects, such as anger, anxiety, irritability, emotional instability, etc. Conscientiousness has negative correlations with all negative mood states and positive correlations with all positive mood states, which is similar to the findings in [21] where both Conscientiousness and Neuroticism showed significant correlation with all mood states. Contrary to [21], Extraversion is positively correlated with Confusion in our observations.

Furthermore, personality traits are found to be associated with mood states as shown in Fig. 4-9 and Fig. Fig. 4-10, illustrated by the comparison between Type 2 and Type 3 persons, whose major differences in personality traits are Neuroticism and Conscientiousness. With lower Neuroticism and higher Conscientiousness, Type 2 individuals have the highest positive and lowest negative mood states while Type 3 persons have the lowest positive mood states.

The influence of mood states and personality traits on driving style is also reflected in Table III; the highest proportion of aggressive drivers is observed in Type 3 personalities, who have the lowest positive mood states. A similar result was presented in [19], indicating that individuals with negative emotions and stress often drove riskily and fast whilst having a low Conscientiousness score. Unlike the big difference between positive and negative mood states seen for Types 2 and 3, each mood state score is neutral for Type 1 individuals and personality traits are close to the mean value (50 in T-score). With balanced mood states and average personality traits, the proportion of sedate drivers for Type 1 is the highest. However, it should be noted that these results do not indicate a strong relationship between personality traits and driving style, in line with [21]. According to Table IV, during the four visits, most of the participants from all three personality types demonstrated more than one driving style. This contradicts a past study on the personality-driving style relationship [113], which finds that driving styles tend to be stable.

In this study, the analysis found that mood states were insufficient in the attainment of a good prediction result for personality type, even though strong correlations were found between the two. Additionally, driver behaviors for the three personality types varied under

different mood states, so driving style prediction using just mood states is also insufficient. Further, solely using either driving style or personality traits achieved better accuracy than using only mood states, but both driving style and personality type predictions can be improved significantly by combining the remaining two types of data as the input.

Limitations. While this study identified the possible prediction of personality traits and driving styles using mood states, refining the predictive model can help offer better personalization. As the study is limited to male drivers, a lack of sufficient demographic representation could constrain the predictive power.

Besides the sample size issue, mood state assessment relies on a subjective self-report, which might not reflect the real mood state of all participants. As we can see, the correlations between agreeableness and all mood states are weak. In some extreme cases, an agreeable person might not report their feelings but may still experience it. As an alternative method, mood states can be objectively assessed by smart devices.

4.2.1.6 Conclusions

In this section, a comprehensive framework that considers driver reaction in each predefined event for the identification of driving style was proposed. By synchronizing the driving style data with mood states, and personality traits data, an integrated dataset was developed. Based on the dataset, the correlation between personality traits and mood states was discussed. This found that Neuroticism had the strongest negative correlation with positive mood states while Conscientiousness had the most negative correlation with negative mood states. Three personality types were determined based on clustering, and it was discovered that Type 1 personalities had the most average personality traits and mood

states. They also demonstrated more sedate driving than the other types. Meanwhile, Type 2 personalities had the highest positive mood states and completed more keen sessions. Type 3 personalities had the lowest positive mood states and drove more aggressively. A prediction model was trained based on random forest and validated, showing that (1) driving style can be predicted using mood states and personality traits and (2) personality types can be predicted using driving style and mood states.

As one of the first few research projects looking into driving style, mood states, and personality traits combined, improvements can be made alongside its continued development. Future studies would incorporate different demographics to evaluate the predictive model for different population groups. On top of that, it is necessary to examine the proposed models in naturalistic environments to assess their stability and usability. As we have already modeled the general driver profile, a personalized driver profile will be considered in the next step. For this, the Myers-Briggs Type Indicator (MBTI) personality assessment, which requires more data from drivers, can be adopted to define personality types.

4.2.2 Personalized Vehicular Interaction Pattern Analysis

4.2.2.1 Introduction and Background

4.2.2.1.1 Motivation

Personalized vehicular interaction pattern analysis reveals a driver's intention to impact their environment and demonstrates how they respond to surrounding conditions, which involves understanding and predicting an individual vehicle's behavior on the road based on historical data. Also, the prediction and decision-making process can be improved by

discovering and encoding the interaction pattern among vehicles to the algorithm, since modeling the interaction enables IV to better coordinate with its surroundings in a safer and more efficient manner.

The increasing amount of vehicle trajectory data has led to data-driven methods for automatically learning the interaction. Most existing studies encode and learn the interaction among agents implicitly as a middle layer of neural networks[114]–[116], which undermines the interpretability and transferability of the learned interaction model. In this study, we aim to model interaction explicitly and address the questions of who is involved in the interaction, when does the interaction occur, and how to quantify it, with an interpretable data-driven model. To identify the complex and implicit vehicle interaction, this study relies on the definition proposed by Markkula et al. [117], who indicate that in an interaction, two or more vehicles should be involved, should be influenced by each other, and should have spatiotemporal conflicts with other vehicles.

Due to its interpretability, Granger causality (GC) [118] is a practical method to analyze interactions on a set of time series, especially for systems with nonlinear dynamics. Recently, it is widely adopted in many fields, including neuroscience, social media analysis, climate science, and econometrics [119], [120]. Similarly, exploring the interaction among vehicles based on the trajectory can be considered a problem of multivariate time series analysis. Therefore, in this dissertation, a GC-based approach is used to explore the interactions among a multi-vehicles system.

In this section, we claim the following key contributions:

- We propose an unsupervised data-driven approach to model the multi-vehicle interaction, as one of the first research that implements Granger causality on a vehicular motion study;
- Regularized by social norms and road geometry, the proposed explainable network is able to quantify the interaction and can be validated in a reliable manner; and
- Besides the demonstration on the INTERACTION dataset, the proposed algorithm is implemented on a personalized dataset and discovers personalized interaction patterns for two drivers.

4.2.2.2 Methodology

In this study, we propose a framework to evaluate the multi-vehicle interaction, which consists of a vehicle behavior model based on potential field theory and a GC discovery process. Similar to the vehicle interaction, movement ecology also considers complicated perception, planning, and execution process. By integrating a conceptual ecology behavioral model [121], Fujii et al. [122] proposed a GC-based inference framework to study the multi-animal interaction, and the performance for GC discovery was validated in the labeled synthetic dataset (e.g., nonlinear oscillator and boid model simulation) and real-world animal dataset. But vehicle movement is more constrained by the social norms (safe behavior) of human drivers and road geometry. To include scientific knowledge regularization, we apply potential field theory in Frenet coordinate [123] to encode the social norms. Overall, as shown in Fig. 4-13, the system inputs are the trajectories of all analyzed vehicles, and the two outputs are the coefficient matrices of GC values for

interaction intensity evaluation and prediction results of each vehicle provided by generalized vector autoregression (GVAR).

Finally, the interaction intensity is quantified by the strength of GC effect in the coefficient matrix, as they explain how the past of other vehicles contributes to the future of the evaluated vehicle.

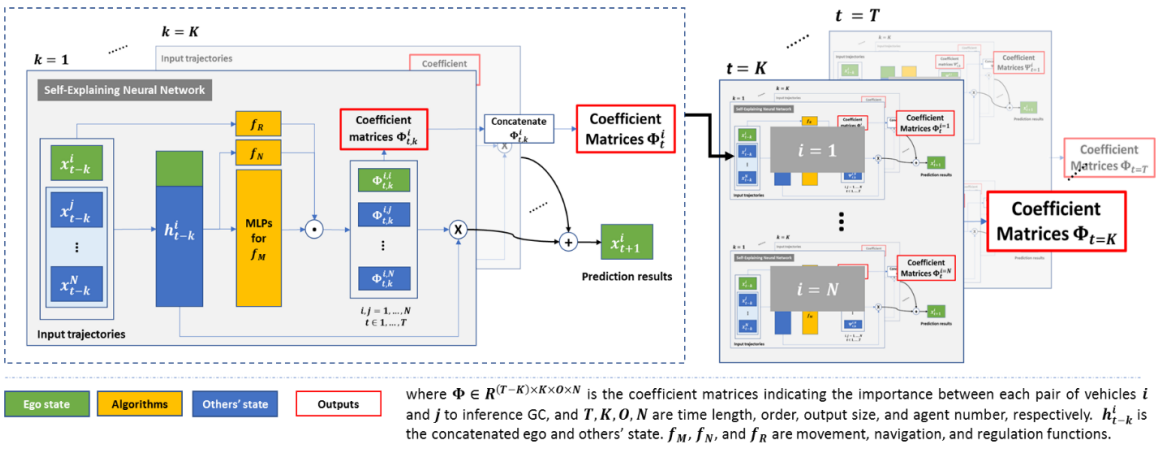


Fig. 4-13 Generalized vector autoregression under self-explaining neural network structure for Granger causality discovery

4.2.2.2.1 GC discovery using GVAR under SENN structure

4.2.2.2.1.1 Granger causality

Granger causality was developed in 1969 by Granger [118] and now became one of the most popular approaches for temporal causal discovery. This dissertation adopted its general form for non-linear systems, and many existing methods can be adapted into this form, such as vector autoregression (VAR) [124] and deep learning methods [122], [125]. Consider observing trajectories from N vehicles, and each trajectory contains P features f across time span $t = \{1, \dots, T\}$, i.e., $X = \{x^1_{<t}, \dots, x^N_{<t}\}, x^i_t = \{f_1, \dots, f_P\}$. A non-linear

function g^i (e.g., nonlinear VAR) is used to capture how the past of all N vehicles influence the i^{th}vehicle, such that

$$x_{t+1}^i = g_i(x_{<t}^1, \dots, x_{<t}^N) + e_t^i \quad (5)$$

where e_t^i is noise term. If g^i is independent on other vehicle $x_{<t}^j$, vehicle $x_{<t}^j$ is irrelevant in the prediction of vehicle x^i. Then, the above can be concluded that [120] time series x^j is noncausal for time series x^i, if and only if for all $\{x_{<t}^1, \dots, x_{<t}^N\}$ and all $x_{<t}^{j'} \neq x_{<t}^j$, such that

$$g^i\left(x_{<t}^1, \dots, x_{<t}^j, \dots, x_{<t}^N\right) = g^i\left(x_{<t}^1, \dots, x_{<t}^{j'}, \dots x_{<t}^p\right) \quad (6)$$

However, false causality may be discovered by only relying on data-driven GC. To improve the reliability of the model, expert knowledge of the study field needs to be integrated into the model. Moreover, instead of defining the system mechanism [126], GC is usually used to investigate complex systems that are difficult to model and to provide a system-level perspective of the interaction.

4.2.2.2.1.2 Self-explaining neural networks (SENN)

A SENN [127] represents a class of intrinsically interpretable models, and it consists of a link function $G(\cdot)$ and interpretable basis $h(x)$, following the form:

$$f(\boldsymbol{x}) = G(\theta(\boldsymbol{x})_1 h(\boldsymbol{x})_1, \dots, \theta(\boldsymbol{x})_k h(\boldsymbol{x})_k) \quad (7)$$

where x are predictors, $\theta(\cdot)$ is a neural network with k outputs. $\theta(x)$ is the coefficient for x and is used to explain the contribution of each basis to prediction result $f(x)$. After simplification, Equation 7 can be written as:

$$f(\boldsymbol{x}) = \sum_{j=1}^{p} \theta(\boldsymbol{x})_j x_j \tag{8}$$

4.2.2.2.1.3 Generalized vector autoregression

In a vehicle interaction study, the trajectory of each involved vehicle is considered as a basis to predict the target vehicle, and the contribution of each basis 'explains' the influence of each vehicle, which is indicated by $\theta(x)$. SENN was applied to infer GC in a multivariate system by Marcinkevics and Vogt [124], who extended VAR to generalized vector autoregression (GVAR) as the link function, and the GVAR is expressed as

$$\mathbf{x}_t = \sum_{k=1}^{K} \boldsymbol{\Phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_{t-k})\mathbf{x}_{t-k} + \varepsilon_t \tag{9}$$

where the $\boldsymbol{\Phi}_{\boldsymbol{\theta}_k}$ is a neural network parameterized by $\boldsymbol{\theta}_k$. Then $\boldsymbol{\Phi}_{t,k}^{i} = \boldsymbol{\Phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_{t-k})$ is a coefficient matrix for lag k and time t, and the element (i, j) of $\boldsymbol{\Phi}_{t,k}^{i}$ is the influence of time step x_{k-t}^{j} on x_t^i.

The dash-box in Fig. 4-13 illustrates the calculation for the coefficient matrix $\boldsymbol{\Phi}_t^i$ and prediction result using Equation 9 with an order- K GVAR. The matrix $\boldsymbol{\Phi}_t^i$ captures the influence of other vehicle on vehicle i at time t, with $\boldsymbol{f}_R$, $\boldsymbol{f}_N$, and $\boldsymbol{f}_M$ representing the regulation, navigation, and movement functions, respectively. These functions are introduced in the following subsections. To obtain the whole coefficient matrix for the whole group, the generalized coefficient matrix of N vehicles are concatenated at each time step.

Therefore, the strength of the influence (i.e., GC effect [122]) between time series x^i and x^j can be explored by inspecting the generalized coefficient matrix $\boldsymbol{\Phi}_t = \boldsymbol{\Phi}_{\boldsymbol{\theta}_k}(x_t)$, as Equation 10:

$$GC^{i,j} = \underset{\substack{1\le k\le K\\ K+1\le t\le T}}{signmax}\left\{\text{median}_{\substack{1\le k\le K\\ K+1\le t\le T}}\left(\boldsymbol{\Phi}_{\boldsymbol{\theta}_k}(\boldsymbol{x}_t)\right)_{i,j}\right\}\cdot \underset{1\le k\le K}{max}\left\{\text{median}_{K+1\le t\le T}\left(\left|\left(\boldsymbol{\Phi}_{\boldsymbol{\theta}_k}(\boldsymbol{x}_t)\right)_{i,j}\right|\right)\right\} \quad (10)$$

where $signmax$ function outputs the sign of the number that has the largest absolute value.

4.2.2.2.1.4 Vehicle movement model

To avoid problematic results (e.g., linking vehicles that are too far apart) for vehicle interaction, the aforementioned knowledge (i.e., social norms) needs to be incorporated into the SENN-GVAR model. Instead of relying on one neural network $\boldsymbol{\Phi}_{\boldsymbol{\theta}_k}$ in Equation 9, we decompose the vehicle movement model into three processes to make the movement model interpretable and reliable, including social norm regularization, navigation, and planning processes (i.e., $\boldsymbol{f}_R$, $\boldsymbol{f}_N$, and $\boldsymbol{f}_M$, respectively, as the yellow blocks in Fig. 4-13 shows). Thus, the $\boldsymbol{\Phi}_{\boldsymbol{\theta}_k}$ in Equation 9 is extended to Equation 11. This subsection discusses how each process in the vehicle movement model is formulated and integrated into the GVAR model.

$$\boldsymbol{\Phi}_{\boldsymbol{\theta}_k}(\mathbf{x}_{t-k}) = f_{R_k}(\boldsymbol{x}_{t-k}) \odot f_{N_k}(\boldsymbol{x}_{t-k}) \odot f_{M_k}(\boldsymbol{x}_{t-k}) \quad (11)$$

A. Regularization Process. The social regularization $\boldsymbol{f}_R$ is formulated to estimate the conflict level with surrounding vehicles based on the region of interest (RoI) of the analyzed vehicle. Driving safety is a critical factor in interaction studies, and drivers are assumed to pay more attention to the conflicting vehicles that may pose a threat. Since the

risk cannot be simply evaluated by Euclidean distance, the potential field becomes a powerful tool for social norm encoding [35] and understanding how drivers perceive their surroundings. To characterize the driving risk level for ego vehicle, Li et al. [128] proposed a potential field equation for vehicle control, whose parameters were calibrated with real-world data, as shown in Equation 12. In this dissertation, the value of risk level is adapted to estimate the RoI of the analyzed vehicle, by proportionally expanding the field based on the 3-second rules [129].

$$\begin{cases} E_v = M_i \rho \frac{e^{-\beta_1 a\cos\, o_0}}{|k'|} \cdot \frac{k'}{|k'|} \\ M_i = m_i(1.566 \times 10^{-14} v^{6.687} + 0.3345) \\ |k'| = \sqrt{\left[(x^* - x_0)\frac{\tau}{e^{\alpha v}}\right]^2 + [(y^* - y_0)\tau]^2} \\ \begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} \cos\,\phi & \sin\,\phi \\ -\sin\,\phi & \cos\,\phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \end{cases} \tag{12}$$

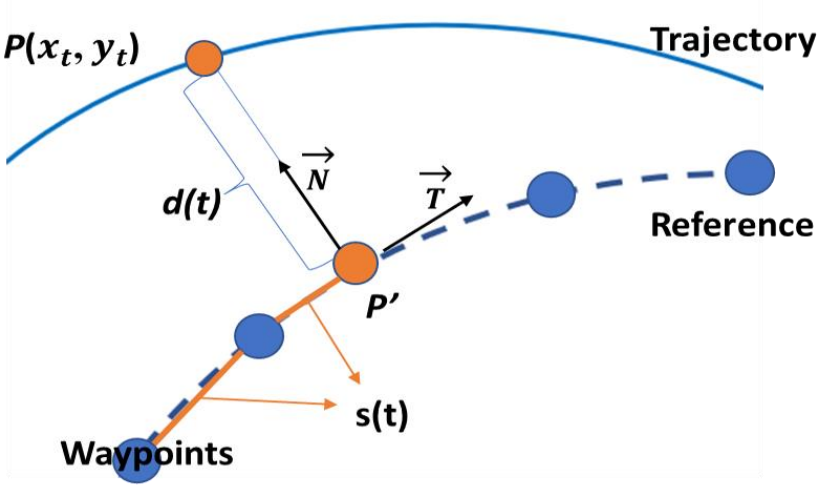


Fig. 4-14 Mapping object in Cartesian coordinate to Frenet coordinate.

***Algorithm 2*: Conflict Level Estimation using Potential Field**

Input: (1) The reference lane. (2) State of all vehicles.

Output: The conflicting levels $\boldsymbol{f_R^{j\to i}}$ of others to analyzed vehicle

1: Convert Cartesian (x_t, y_t) to Frenet frame (s_t, d_t) for the vehicle state of the analyzed vehicle

2: Create a grid map $G(s,d)$ centered on (s_t, d_t)

3: Calculate risk value $E(s,d)$ for the grid based on Equation (8)

4: Find the contour S_c of 95$^{\text{th}}$-percentile (2σ) of $E(s,d)$

5: Calculate ratio $r = 3 \cdot v/max(S_c)$, where $3 \cdot v$ follows the three-second rule

6: Expand the grid map $G'(s',d') = G(s,d) \cdot r$, and obtain the $E(s',d')$

7: Convert $E(s',d')$ in Frenet frame to $\boldsymbol{E(x,y)}$ Cartesian frame

8: Calculate the risk level $\boldsymbol{R^{j\to i}}$ of other vehicle V^j, based on the position $P^j(x,y)$ on $\boldsymbol{E(x,y)}$

9. $\boldsymbol{f_R^{j\to i} = R^{j\to i}/max(E(x,y))}$.

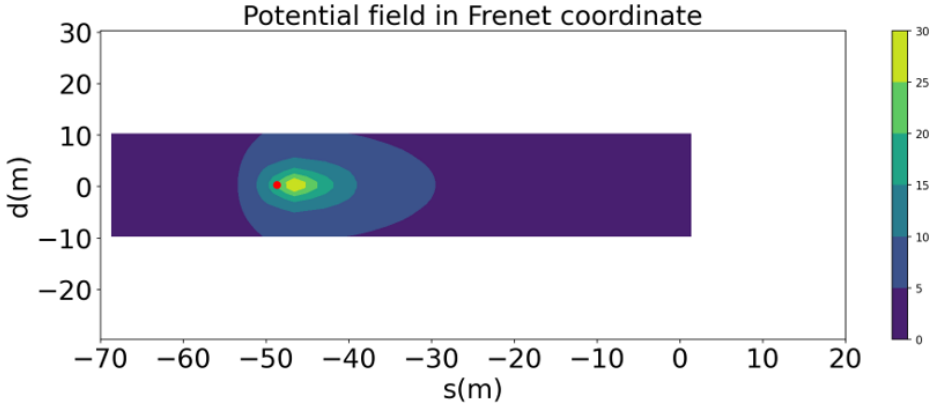
Return: $\boldsymbol{f_R^{j\to i}}, \forall j = 1, \dots, N, j \neq i$

Besides the social norm, road geometry needs to be included. Therefore, we calculate the RoI in Frenet coordinate, instead of Cartesian coordinate. As shown in

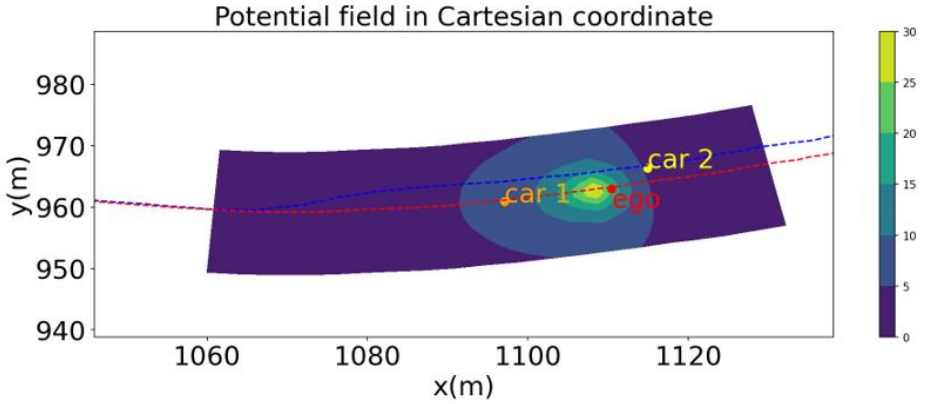
Fig. 4-14, vehicle state (e.g., longitudinal position $s(t)$ and lateral position $d(t)$) in Frenet coordinate is calculated based on the reference lane (e.g., the center line of the lane). At each time step, Algorithm 2 generates an RoI for each vehicle as shown in Fig. 4-15, where the red dot stands for the analyzed vehicle, and blue dash lines and red dash lines are the reference lanes. Based on the RoI, the conflict level of other vehicles can be estimated by considering road geometry and the analyzed vehicle state.

B. Navigation and Movement Process. The navigation process $\boldsymbol{f_N}$ is a sign function to capture the car-following behavior between ego vehicle i and vehicle j, and $f_N(x^{i,j}) =$

$2\sigma\left(\Delta v^{i,j}/w_n\right) - 1$, where σ is a sigmoid function, $\Delta v^{i,j}$ is the speed difference, and w_n is a learnable coefficient. $\boldsymbol{f}_N > 0$ indicates attraction, i.e., the vehicle tries to close the car-following gap, and $\boldsymbol{f}_N < 0$ means repulsion, i.e., the opening gap behavior. The movement process f_M^i is constructed by two-layer MLPs for each order and vehicle.



(a) RoI of the analyzed car in Frenet frame



(b) Conflict level of other vehicles within RoI

Fig. 4-15 An example of estimating the conflict level of surroundings based on potential field theory

The loss function for training the whole GVAR consists of mean square prediction error (MSE), sparsity-inducing penalty [130] (i.e., $\mathcal{L}_s = \alpha\|\Phi_t\|_1 + (1-\alpha)\|\Phi_t\|_F^2$), and a theory-guided regularization term $\mathcal{L}_{TG}$. $\mathcal{L}_{TG}$ is adopted with the assumption that the vehicle

goes straight following the center line from the current state if no interaction is detected, as shown in Equation 13. The whole loss function can be written by Equation 14.

$$\mathcal{L}_{TG}(\boldsymbol{\Phi}_t) = \exp\left(\|\boldsymbol{x}_t - \widetilde{\boldsymbol{x}}_t\|_2^2/\eta\right)\|\Phi_t'\|_t^2 \tag{13}$$

$$\mathcal{L} = \frac{1}{T-K}\sum_{t=K+1}^{T}\left(\|x_t - \widehat{\boldsymbol{x}}_t\|_2^2 + \lambda\mathcal{L}_s + \gamma\mathcal{L}_{TG}\right) \tag{14}$$

C. Permutation Feature Importance for Causality Validation. The cause discovered by GC is the potential cause, which needs to be validated. Besides the visualization of GC value, we use permutation feature importance (PFI) [109] to validate the proposed data-driven method without ground truth for interaction. PFI measures a method for determining the importance of a feature in a machine learning model, and similarly, it is used to understand the contribution of a vehicle to the overall prediction. It works by randomly shuffling the values of the trajectory of the target vehicle, and measuring the change in the model's performance.

On the other hand, PFI is a powerful technique to validate the result of GC. Since causality discovery relies on temporal information, the permutation process can remove chronological information and causal relation between the target vehicle and the rest of the system, before re-sending the data into the GC-based network. Also, PFI introduces no confounding factor to affect the prediction, as permutation does not change the distribution of the dataset. The validation process based on PFI is described in Algorithm 2.

Algorithm 2: **Vehicle Contribution Analysis based on PFI**

Input: (1) Trained network model. (2) Input dataset $\boldsymbol{X}$.
Output: An importance matrix $\boldsymbol{M_{PFI}}$ for $\boldsymbol{N}$ vehicles
Create $\boldsymbol{M_{PFI}} \in \boldsymbol{R^N}$
Run the trained network on dataset $\boldsymbol{X}$
Measure the loss $\boldsymbol{L_o}$
for each vehicle x^i in system:
    Create $\boldsymbol{X_{perm}}$ by permuting features of vehicle x^i
    Run the trained network on the $\boldsymbol{X_{perm}}$
    Measure the intervention prediction loss $\boldsymbol{L_p}^i$ of x^i
    Calculate importance as quotient $\boldsymbol{FI}^i = \boldsymbol{L_p}^i/\boldsymbol{L_o}$
    $\boldsymbol{M_{PFI}(i)} = \boldsymbol{FI}^i$
Return: $\boldsymbol{M_{PFI}}$

4.2.2.3 Case Study and Result Analysis

In order to validate the proposed GC-based vehicle interaction modeling approach, we studied the vehicle interaction in a real-world on-ramp merging scenario, which requires the coordination of lateral and longitudinal control from drivers, making it a highly interactive and conflicting scenario. In this section, the proposed algorithm reveals how the studied vehicle influences others and is influenced in a four-vehicle merging scenario. Moreover, the algorithm recognizes the personalized interaction patterns of two drivers with different driving styles, using their historical datasets.

4.2.2.3.1 Vehicular interaction interpretation using INTERACTION dataset

Among various datasets, we demonstrate the proposed algorithm on the INTERACTION dataset [131], which provides HD maps and motions of all vehicles which may influence driving behavior. Trajectories of each vehicle include the timestamp, position (x, y), speed (v_x, v_y), and heading angle. The selected ramp merging scenario is presented in Fig. 4-16,

where the center lines of the ramp and mainline are extracted from the HD map, and a four-vehicle group is chosen as the study case.

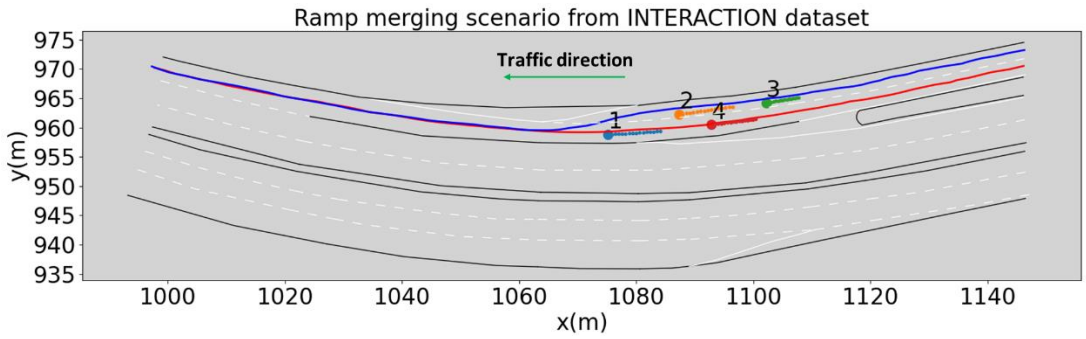


Fig. 4-16 On-ramp merging scenario from INTERACTION dataset.

Once $\boldsymbol{\Phi}_{\boldsymbol{\theta}_k}$ is trained, the vehicular interaction intensity ($GC^{i,j}$) can be assessed by Equation 10. The four-vehicle interaction during the merging process is elaborated in Fig. 4-17. Fig. 4-17(a) presents four key events during the merging, where the blue and red dash lines are the ramp and mainline, respectively. Vehicles are numbered based on their appearance order with tails of past 1.5 seconds (15-time steps), and a longer tail implies higher speed. The $GC^{i,j}$ throughout time is described in Fig. 4-17(b), e.g., (i,j) graph is the influence of vehicle # i on vehicle # j. A positive value means attraction and following, while a negative value stands for repulsion and implies that rather than reacting to another's action, the agent tends to take the lead. Moreover, in each graph, four vertical dash lines mark the four key events in Fig. 4-17(a) correspondingly.

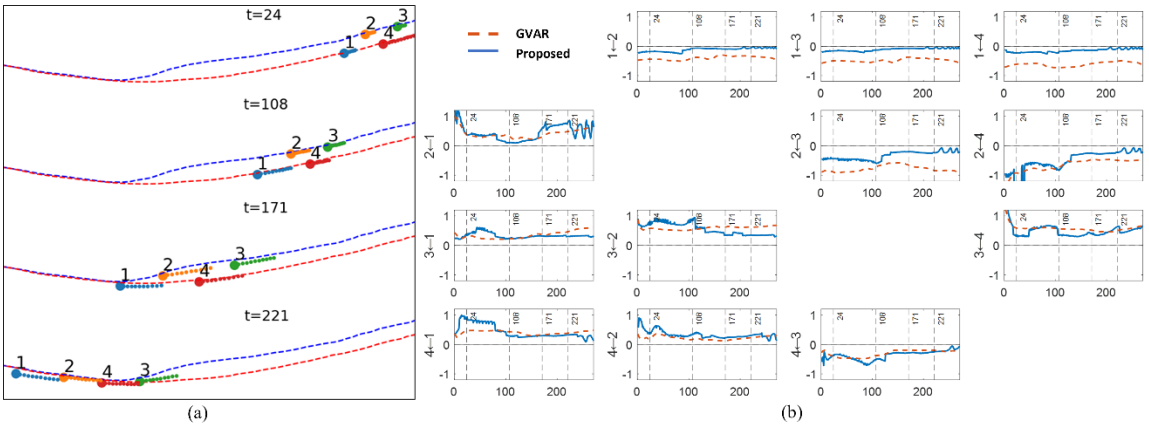


Fig. 4-17 Interaction process during a four-vehicle merging. (a) Key moments during the merging process. (b) Interaction intensity among vehicles.

The blue solid line in Fig. 4-17(b) shows that the proposed algorithm can detect the most interactive at each time step for each vehicle. At t =24, vehicle #4 joins the group with a higher speed (longer tail). According to graph (4,1) in Fig. 4-17(b), #4 is significantly affected by #1 since it has to slow down and follows #1. Also, #4 competes for the merging priority with #3 and tries to lead #3, and this movement is reflected on both graph (4,3) and (3,4). At the same time, #2 is taking the lead over #4 and shows a strong will to compete. At t =108, since #1 speeds up and leaves everyone behind, its influence on others decreases, and others have little influence on #1. At the moment, #2 takes #4 as the most important competitor and accelerates, while most of #4's attention is on #3. When t =171, the competition of merging order is settled, so the interaction intensity among vehicles becomes more stable, except for #2, which starts merging into the mainline and needs to maintain a safe car-following gap to #1. At t =221, the last vehicle #3 finishes its merging. The answer to who is involved in interaction and when the interaction happens can be solved by setting an intensity threshold, based on algorithms that distinguish the difference between non-interactive state and interactive state. Besides rule-based methods, there are

many algorithms that can be used for unsupervised classification, to name a few, clustering methods (e.g., KNN and DBSCAN), volatility analysis and Otus's method.

As the baseline, GVAR has no information about the road constraint and social norms in traffic and generates counterintuitive results, as the orange dash line in Fig. 4-17(b) shows. For example, GVAR misjudges that the far upstream vehicles affect #1, as in graph (1,3) (1,4) (2,3). In addition, although #3 increases the car-following gap to #2 and switches to follow #4 at t =108, the effect of #2 on #3 is maintained till the end by GVAR, as in graph (3,2). Moreover, GVAR ignores the perception limitation of #3 and identifies an increasing effect of #1 on #3 at the end of the graph (3,1), even though there are two vehicles in between.

TABLE VI PERMUTATION FEATURE IMPORTANCE (PFI) ANALYSIS

Vehicles	#1	#2	#3	#4
Influence Level *	312.18	327.81	390.99	434.45
Final Epoch Loss	1.39			
Permutation Loss	1.02	1.27	1.73	2.04
Quotient - PFI	0.66	0.91	1.25	1.46

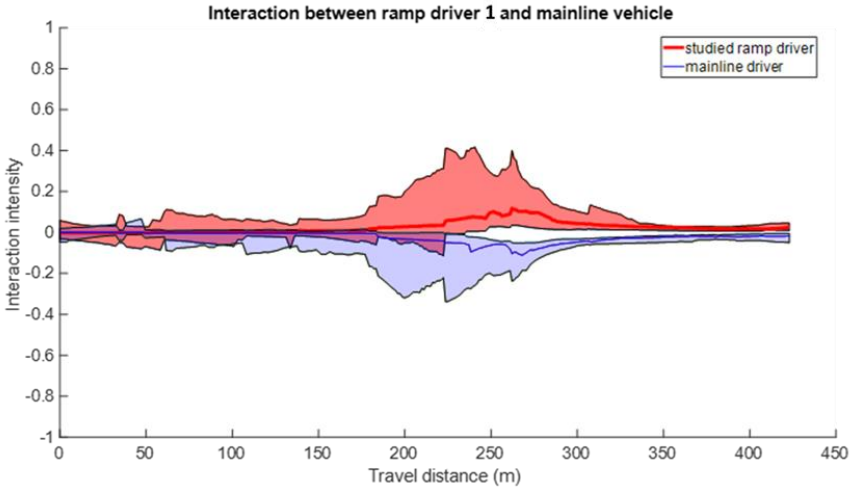
*A vehicle's overall influence level is measured by the sum of its GC value over time and GC values of other agents, i.e., the influence level of x^i is $\sum_{j\in N, j\neq i}^{N}\sum_{t}^{T}\left|GC^{ij}\right|$

The contribution of each vehicle to the system prediction is indicated by its PFI value, as listed in Table VI. A vehicle's PFI value aligns with its GC value. In other words, if the influence (measured by its GC) level is higher over time, its PFI is higher. For instance, #1 contributes the least, so both values of its GC and PFI are low. On the other hand, the PFI values of #4 are the highest since it interacts with both #2 and #3. It should be noted that a

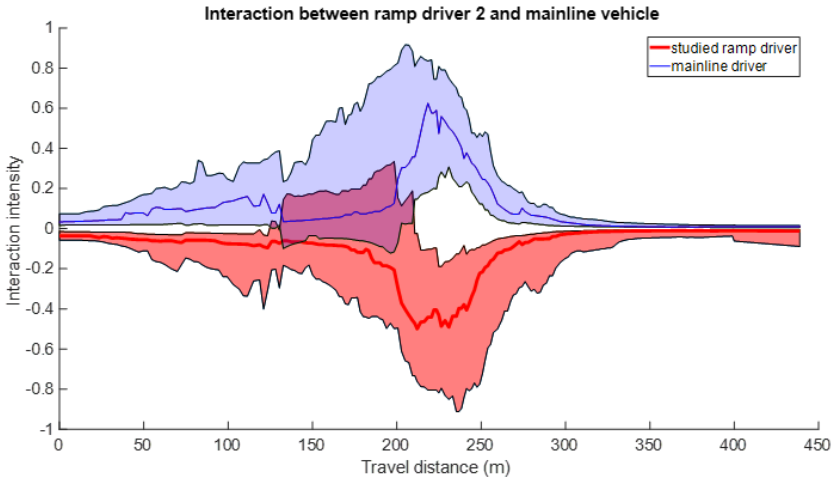
vehicle's GC value does not include its contribution to its own prediction, therefore, there is no linear mapping relationship between GC and PFI.

4.2.2.3.2 Personalized interaction pattern discovery

Understanding how individuals interact with other vehicles and the interactions differ from each other allows engineers to design personalized vehicles that better meet the needs and preferences of different drivers. To study the personalized interaction behavior, we implement the proposed algorithm on a personalized dataset of two drivers from our previous study [132]. The dataset was collected in ramp merging field experiments that are carried out at an on/off-ramp section along Columbia Ave., Riverside, CA. The experiment created merge interactions between a ramp vehicle and a mainline vehicle and collected 20 merging trips for each studied ramp driver and an anonymous mainline driver. In the driver behavior analysis, Driver 1 was found to be more conservative and usually merged behind the mainline vehicle. In contrast, Driver 2 was more aggressive and always accelerated to merge in front of the mainline vehicle.



(a) Aggregated interaction pattern of conservative ramp car driver 1



(b) Aggregated interaction pattern of aggressive ramp car driver 2

Fig. 4-18 Personalized interaction pattern analysis in ramp merging scenario

Fig. 4-18 illustrates the personalized interaction pattern during ramp merging trips for these two drivers. The red line is the average interaction intensity for the studied ramp driver, and the blue line is for the conflicting mainline driver. For conservative Driver 1 in Fig. 4-18(a), the interaction between two drivers is mild, with a maximum intensity value of 0.41. The positive intensity value of Driver 1 shows an attraction to the mainline vehicle, which leads to the movement of Driver 1. Compared to Driver 1, Driver 2 involves stronger

interactions in Fig. 4-18(b), where the maximum absolute intensity is 0.92, and the intensity variance is much larger than Driver 1's. For most of the time, the interaction intensity of Driver 2 is negative, since Driver 2 prefers to take the lead and merge in front.

4.2.2.4 Discussion

Implications. The goal of this study is to provide an interpretable data-driven method to discover the interaction pattern from observed vehicle trajectories. As a result, the mutual influences among vehicles are quantified by the strength of the GC effect, and we are able to answer the unsettled questions of who, when, and how for interaction study. The proposed method in this dissertation lays a solid foundation for a suite of downstream applications, including multi-vehicle trajectory prediction, traffic organization, and motion planning. For example, this method can be used as an automatic data labeling tool to encode interaction information into the neural networks for prediction purposes. Also, if the interaction intensity is too high, traffic control (e.g., ramp metering) can be implemented to relieve the competition among vehicles considering safety. Finally, personalized interaction can be studied for specific drivers by following this protocol. Knowing the personalized interaction preference can help customize the vehicle setup for a personalized advanced driver assistance system, enabling the vehicle to interact with others in dynamic environments. Moreover, modeling personalized interaction contributes to driving style recognition and personality inference [133].

Limitations. One obvious shortcoming of using GC is that the VAR-based prediction methods cannot capture long-term temporal dependency whenever GC takes advantage of VAR's simple structure. Similar to other unsupervised methods, validating the interaction

without ground truth is still challenging. Toward this end, Granger causality is not the real cause, and the result may not be the only cause. However, it can be used as an analytical tool to reveal the interaction within a system.

4.2.2.5 Conclusions

In this dissertation, we quantify the multi-vehicle interaction using an explainable data-driven approach, which is one of the first implementations of Granger causality on vehicle motion study. To improve accuracy and interpretability, the proposed approach integrates social norms and road geometry into the network. An on-ramp merging scenario with real-world data is used to demonstrate the algorithm performance, and permutation feature importance is used to validate the result for this unsupervised algorithm. Finally, the algorithm is implemented on a personalized driving dataset for personalized interaction pattern recognition.

As the first few GC implementations in the vehicle movement domain, there are many future directions worth exploring. For example, the interaction between vehicles and infrastructure (e.g., traffic lights and traffic signs) can be considered since the movement of vehicles is also influenced by static objects on the road. We will use the proposed method as a data labeling tool for network training to enable interaction-aware prediction. Discovering personalized interaction patterns for P-ADAS development can be also one of our future studies.

4.2.3 Personalized Driving Behavior Modeling with Inverse Reinforcement Learning

4.2.3.1 Introduction and Background

To perform motion planning in complex mixed-traffic, CAVs need to predict human-driven vehicles' behaviors, make decisions in response to the actions (in presence or to be taken), and execute the right maneuvers through the planner and controller. Particularly, the prediction of human-driven vehicles' behaviors is challenging due to the uncertainties of human drivers. As a result, many researchers [11], [133]–[135] include driving behavior modeling in the prediction and planning stage, where driver type classification and identification play an important role. Nevertheless, the improvement of integrating the collective driving behavior is limited, because driving behavior can be diverse among different drivers.

After studying vehicular interaction, we can quantify the influence among vehicles and find out the most critical surrounding agents. However, the reasoning process behind interaction is more complex, considering the varied driving preference. To facilitate the motion planning, in addition to the high-level behavior modeling, driving preferences at maneuver level (i.e., longitudinal and lateral behavior) need to be discovered.

To study personalized driving behavior, this section deploys the Driver Digital Twin (DDT) in the real world. DDT is a digital replica of a driver with his or her naturalistic driving data and driving behavior models. Based on real-world data, DDT system in the virtual world provides both online and offline micro-services, e.g., interactive prediction, driving style analysis, etc. In addition, to capture the driving preference variation, the

evolving driver model will be updated in a certain period (e.g., every five new trips) by consuming the driving data from the real-world vehicle.

As a typical example of personalized driver behavior that can be modeled by DDT, lane changing is a fundamental but challenging task in our daily driving, especially in mandatory lane change situations (such as ramp merging) where the open areas are very limited and levels of risk are higher, compared to discretionary lane changes. In these situations, it is particularly important to accurately predict lane change intention as well as when and where the lane change occurs, because the lane change maneuver will have a significant impact on the safety and efficiency of the road network.

In this section, learning-based algorithms for personalized behavior modeling are developed, and field implementations are carried out on a customized vehicle-edge-cloud platform under the digital twin framework. In the field implementation, the cloud (i.e., Amazon Web Services) analyzes the personalized behavior of human-driven vehicles with connectivity (e.g., by cellphone) and stores the learned driver models and historical data.

This section has the following main contributions:

- A hierarchical learning-based system is developed for personalized driving behavior modeling,
- Under the digital twin framework, a vehicle-edge-cloud platform is constructed and demonstrated, enabling real-world data collection and algorithms development.
- Personalized driver models are trained and validated using the vehicle-edge-cloud platform, as one of the first real-world deployments of Driver Digital Twin in transportation.

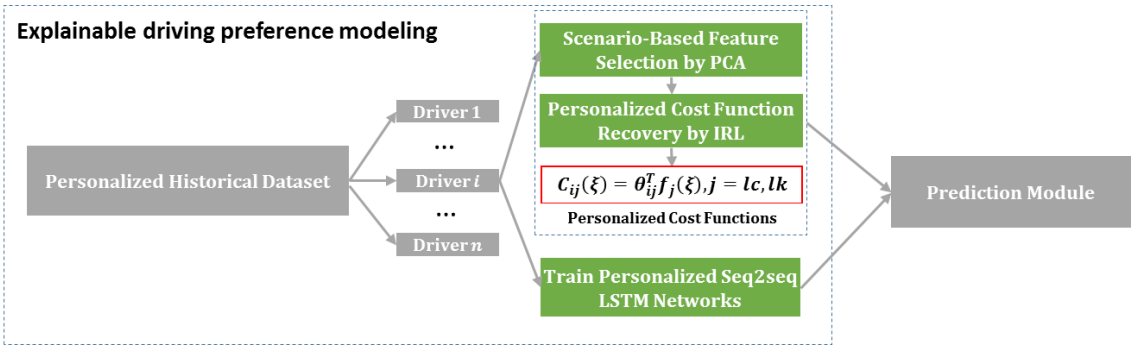


Fig. 4-19 Personalized driving preference modeling by recovering cost function.

4.2.3.2 Methodology

This personalized behavior modeling algorithm is first developed in a simulation environment and then improved to be a real-world implementable version. As shown in Fig. 4-19, the personalized driving behavior for each driver is learned in a offline phase based on the personalized dataset collected from the specific driver. For each driver, a neural network structure sequence-to-sequence (Seq2seq) structure [136] based on LSTM is adopted to predict the lane-change intention, learning the lane change driving behavior implicitly. To model the driver preference explicitly, the cost functions inferring the driver preference are learned by IRL for supporting the prediction. In this section, we focus on the driving preference modeling by IRL, while the implicit driving maneuver modeling by Seq2SeqLSTM will be discussed in Section 4.3.1, along with the prediction.

4.2.3.2.1 Continuous IRL

Driving preference are usually represented by the cost function, and rational drivers are assumed to behave for optimizing their cost functions. Considering the continuity of the trajectory space, this study adopts Continuous IRL with Locally Optimal Examples [13], [137] to recover this unknown cost function from expert demonstrations.

The cost function is a linear combination of a set of features, i.e., $C_i(\theta_i, \xi) = \theta_i^T f_i(\xi), i = a_{\text{change}}, a_{\text{keep}}$, where θ_i^T is the weights vector emphasizing the features, $f_i(\xi) = \|f_i(s_1, s_2, \dots, s_t)\|_2$, and trajectory sequence $\xi = \left(s_1^i, \dots, s_t^i\right)$. The goal of the IRL is to figure out the optimal weights θ_i^* to describe each driver's preference, which maximizes the likelihood of the driver's historical trajectories $\Xi = \{\xi_k\}$, shown in Equation 15:

$$\theta_i^* = \arg \max_{\theta_i} P(\Xi \mid \theta_i) \tag{15}$$

According to the principle of maximum entropy, as shown in Equation 16, a trajectory with a low cost has a higher probability, which is proportional to the exponential of its cost, which is proportional to the exponential of its cost.

$$P(\xi \mid \theta_i) = \frac{e^{-C_i(\theta_i, \xi)}}{Z(\theta)} = \frac{e^{-\theta_i^T f_i(\xi)}}{\int e^{-\theta_i^T f_i(\tilde{\xi})} d\tilde{\xi}} \tag{16}$$

where $Z(\theta) = \int e^{-\theta_i^T f_i(\tilde{\xi})} d\tilde{\xi}$ is the partition function integrating all arbitrary trajectories $\tilde{\xi}$. To handle the computational complexity in solving the partition function, the continuous IRL approximates $C_i(\theta_i, \xi)$ using the second-order Taylor expansion around the demonstrated trajectory ξ, as in Equation 17. As a result, the partition function is now a Gaussian integral and becomes analytically solvable. With combining Equation 15, the problem is reformulated to a minimization of $-\log P(\Xi \mid \theta_i)$, as in Equation 18:

$$C(\tilde{\xi}) \approx C(\xi) + (\tilde{\xi} - \xi)^T \frac{\partial C}{\partial \boldsymbol{u}} + \frac{1}{2}(\tilde{\xi} - \xi)^T \frac{\partial^2 C}{\partial \xi^2}(\tilde{\xi} - \xi) \tag{17}$$

$$\theta_i^* = \arg\min_{\theta_i} \sum_{k=1}^{K} \frac{1}{2}\mathbf{g}_{\theta_i}^T(\xi_k)\mathbf{H}_{\theta_i}^{-1}(\xi_k)\mathbf{g}_{\theta_i}(\xi_k) - \frac{1}{2}\log\left|\mathbf{H}_{\theta_i}(\xi_k)\right| \tag{18}$$

where $\mathbf{g}^T$ and $\mathbf{H}$ are the gradient and Hessian, respectively. This formula indicates that along the expert demonstration, the recovery cost function should have small gradients and large positive Hessians.

4.2.3.2.2 Cost function feature selection

The selected features present the vehicle state in an interpretable way and can capture the preference of the driver. We select the following features to calculate the cost function, based on the available Inertial Measurement Units (IMU) and Global Navigation Satellite Systems (GNSS) information.

a) Car-following risk: the time headway to the leading vehicle,

$$f_{\text{risk}_f} = 1 - \tanh\left(h_{ev}/H_{\min}\right), h_{ev} = d_{\text{headway}}/v_{\text{lon}} \tag{19}$$

where $H_{\min}$ is the minimum safe time headway based on the 3-second rule [129], h_{ev} is the time headway of ego vehicle to leading vehicle, and d_{headway} is the distance to the leading vehicle.

b) Lane-change risk f_{thw}: ego vehicle is projected to its adjacent lane and calculates the time headway to its potential leading vehicle and the time headway from its following vehicle.

$$f_{risk_{lc}} = \frac{1-\tanh\left(h'_{ev}/H_{min}\right)+1-\tanh\left(h'_{fv}/H_{min}\right)}{2} \tag{20}$$

where h'_{ev} is the project time headway of ego vehicle to potential leading vehicle, and h'_{fv} is the time headway from the potential following vehicle.

c) Lane-change urgency f_{urge}: If the ego vehicle needs to perform a mandatory lane change, the remaining time distance should be considered.

$$f_{\text{urge}} = \frac{\left[1+\tanh\left(\frac{L_{\text{widh}}}{2}-y\right)\right]\left[1+\tanh\left(\frac{(X_m-x)}{(v_{\text{lon}}\cdot H_{min})}\right)\right]}{max(f_{\text{urge}})} \tag{21}$$

where the L_{widh} is the width of the lane, X_m is the longitudinal location of the midpoint of merging area, x and y are the locations of ego vehicle, and the $max(f_{\text{urge}})$ is the maximum of f_{urge}, for normalizing the feature.

An example surface of this feature is shown in Fig. 4-20, which illustrates how this feature varies within a 200m lane-change area, with a lane width of 4m. As the vehicle comes closer to the end of the lane-change area without changing the lane, the urgency increases. But once the lane change is completed, the urgency will decrease to zero shortly.

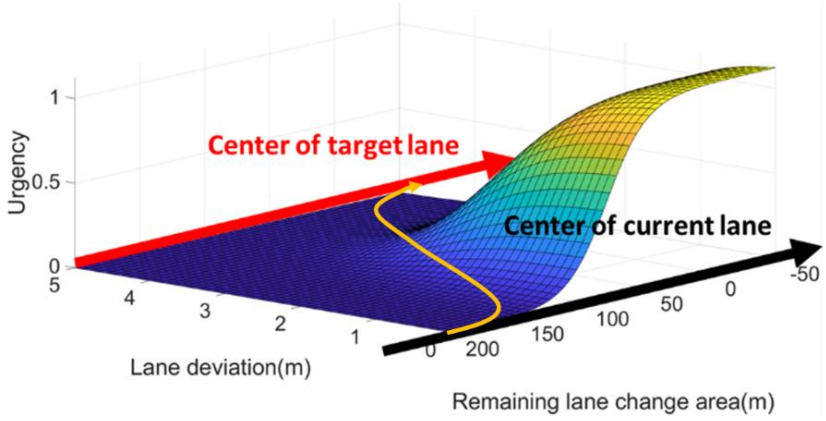


Fig. 4-20 The urgency for a mandatory lane change.

c) Mobility f_m: Drivers have different preferences on mobility. The difference between current speed and the speed limit (v_{lim}) is used to evaluate this preference.

$$f_m = 1 - e^{-(v_{lim} - v_{lon})^2} \tag{22}$$

d) Comfort f_{c1} and f_{c2}: The absolute value of the longitudinal acceleration a_{lon} and the yaw rate ω is used to gauge comfort preference.

$$f_{c1} = |a_{lon}|, f_{c2} = |\omega| \tag{23}$$

e) Lane deviation f_{dev}: We also include lateral distance into the cost function to evaluate the imperfection of driving along the centerline of the lane even in the lane-keeping stage.

$$f_d = |y - Y_c| \tag{24}$$

where Y_c is the location of the centerline of the lane, y is the lateral position of the ego vehicle.

Considering the driver's focus may be different in each scenario (e.g., a driver may care about the lane-change risk and the remaining distance when changing the lane, but not when keeping the lane), we select two groups of features: $\left\{f_{\text{risk } f}, f_{\text{risk } lc}, f_{\text{urge}}, f_m, f_{c1}\right\}$ for lane-change maneuvers, and $\left\{f_{\text{risk } f}, f_{\text{m}}, f_{\text{c1}}, f_{c2}, f_d\right\}$ for lane-keep maneuvers, respectively.

4.2.3.3 Experiments and Results

4.2.3.3.1 Personalized dataset collection

To validate the proposed algorithm in real world, a personalized dataset for each ramp vehicle (RV) driver and its paired anonymous mainline vehicle (MV) drivers is collected, as shown in Fig. 4-21. In total, for each driver, 20 trips entering the mainline and 20 trips

driving off-ramp are used for behavior modeling. The average duration of each trip is 35 seconds, with an average update rate of 5 Hz. In each time step, both the trajectories of mainline vehicle and ramp vehicle are recorded and synchronized at edge server, and the dataset on the cloud server is updated at the end of the trip.

Fig. 4-21 Personalized dataset collection using the Vehicle-Edge-Cloud digital twin platform

4.2.3.3.2 Cost function and driving pattern analysis

Personalized models are trained using drivers' own datasets (i.e., 19 trips collected from Driver 1 and 20 trips from Driver 2). For two drivers, a general model is trained using the aggregated dataset from both drivers, standing for collective driving behavior. The weights in the cost functions can reflect drivers' preference when large weights penalize high values, as shown in Table VII. For lane change maneuvers, Driver 1 cares more about longitudinal comfort and penalizes large acceleration with 0.696 on f_a, compared with 0.425 for Driver 2. Driver 2 prefers a smaller time gap during lane change as he/she puts a larger weight on $f_{\text{risk}\ lc}$ than driver 1 to penalize the time headway. Driver 1 does not care about the mobility f_m and drives slower than the speed limit, while

Driver 2 prefers to drive faster with a weight of 0.182. For lane-keeping maneuvers, two drivers show the same preference for longitudinal comfort f_{c1} but significant differences in lateral comfort and mobility. Driver 1 tends to keep the vehicle stable (i.e., putting a large weight on f_{c2} penalizing unstable yaw movements) and drives slowly. On the contrary, Driver 2 pays little attention to the stability of the vehicle pose and seeks high speed. According to the analysis of cost function weights, Driver 1 is more likely to be a cautious driver while Driver 2 tends to be more aggressive. For lane changing, the general model puts the least weight on urgency f_{urge} and penalizes large time headway. For lane keeping, the general model is sensitive to lane deviation f_{dev} the most.

TABLE VII WEIGHTS OF RECOVERED COST FUNCTIONS

Scene	Driver	f_{c1}	$f_{risk\ lc}$	f_{dev}	f_{c2}	f_m	f_{urge}
LC	#1	0.696	0.151	0	0	0.023	0.126
	#2	0.425	0.240	0	0	0.182	0.151
	General	0.529	0.284	0	0	0.179	0.008
LK	#1	0.388	0	0.246	0.348	0.017	0
	#2	0.323	0	0.222	0.043	0.412	0
	General	0.356	0	0.311	0.309	0.023	0

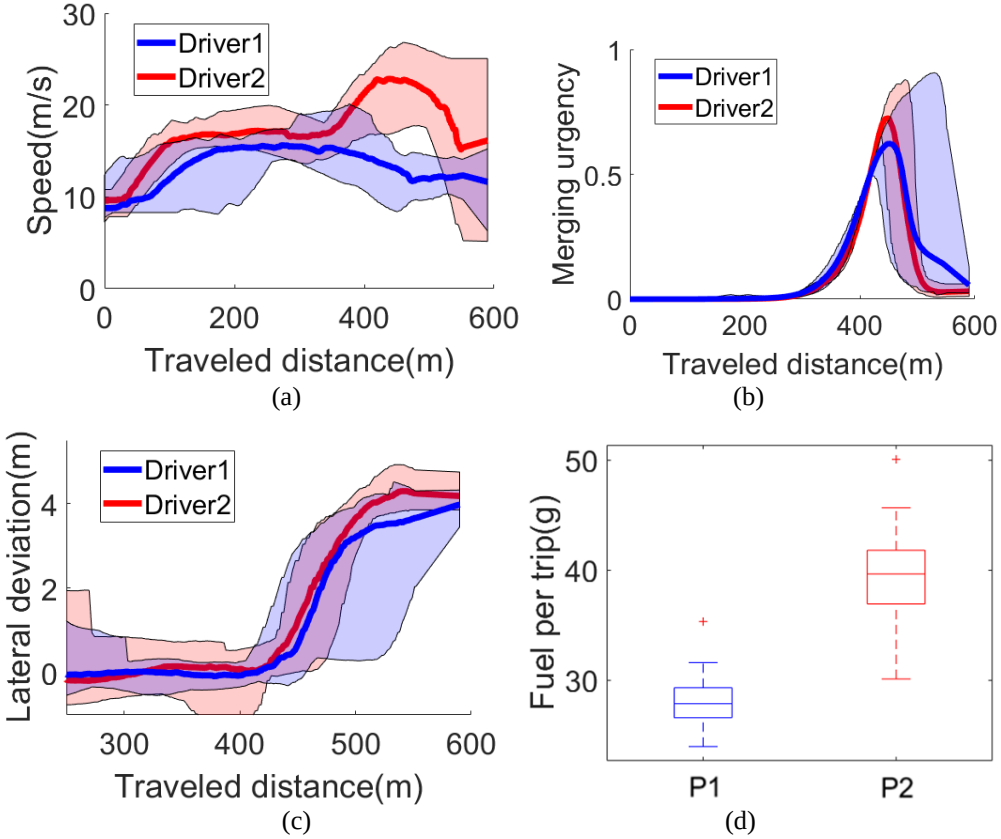


Fig. 4-22 Driving pattern comparison during a lane change process: (a) longitudinal speed of ramp vehicle (b) mandatory lane change urgency (c) lateral behavior and (d) fuel consumption.

Besides the cost function analysis, the driving pattern can also be recognized via the overall lane change behavior, as shown in Fig. 4-22.

The interaction with the mainline vehicle can be reflected by the longitudinal speed of the lane change process in Fig. 4-22(a), which displays the median speed at each location. The observable point (at 320m) is 100 meters before the lane change point, where mainline and ramp drivers can see each other for the first time during the lane change. After observing the conflict with MV, Driver 1 chooses to slow down and yield to MV for lane change behind, but Driver 2 accelerates to surpass in order to cut in front of MV. The lane change urgency in Fig. 4-22(b) is used to measure how the driver deals with a mandatory

lane change. The urgency value grows when RV comes closer to the end of the lane-change area without changing the lane. Once the lane change is completed, the urgency will decrease shortly. Driver 1 has a smaller peak value than Driver 2 for lane change urgency; however, the urgency pattern of Driver 2 is more consistent, as the urgency variation (red area) is smaller than the one of Driver 1 (blue area).

The lateral movement preference is captured by the lateral deviation shown in Fig. 4-22 (c), where the lane change is completed once the lateral deviation reaches 4m or above. Although two drivers have different preferences for lane change sequences, their lane change starting points are close to each other. Moreover, two types of slopes (i.e., lane change speed) in the blue line are observed during the lane change process of the driver. In the first segment, Driver 1 merely crosses the lane separation line, and in the second segment, he/she approaches the center line slowly after confirming safety.

A similar conclusion to cost function analysis can be made that Driver 1 is more cautious than Driver 2. As a result, as shown in Fig. 4-22 (d), the average fuel consumption of Driver 1 (28.2g) is 28.9% less than the one of Driver 2 (39.7g), in each lane change.

4.2.3.4 Conclusions and Future Work

This section introduced a personalized driving behavior modeling algorithm based on inverse reinforcement learning, based on the real-world personalized dataset collected on the vehicle-edge-cloud Digital Twin platform. The recovered cost functions distinguish conservative driver1 and aggressive drive2, aligning with the results of driving pattern analysis. The recovered cost function represents the driving preference of the driver and will be used for the downstream prediction and motion planning module.

4.3 Predictive Behavior-Aware Planning Strategy

In order to navigate complex and dynamic traffic scenarios effectively, IV must possess the ability to anticipate and predict the behavior of other road users. This necessitates the development of a predictive behavior-aware planning strategy, wherein vehicles integrate human behavior into the motion planning module. By incorporating predictive capabilities into the planning process, intelligent vehicles can proactively make decisions that enhance safety, efficiency, and overall driving experience.

In this section, we explore the methodologies underlying the predictive behavior-aware planning strategy, showcasing its potential to transform autonomous driving and foster a more cooperative and harmonious transportation ecosystem. We begin with "Prediction for Driver Intention and Vehicle Trajectories," followed by an examination of "Ramp Merging Strategy with Feedforward Planning and Feedback Control." Lastly, we discuss "Driving Behavior Adaptive Advanced Driving Assistance System," which offers driving style adaptive speed suggestions, highlighting the application of this predictive strategy in practical, real-world scenarios.

4.3.1 Prediction for Driver Intention and Vehicle Trajectories

4.3.1.1 Introduction and Background

Our transportation systems cannot achieve full automation/connectivity in the short future, where CAVs have to interact with human-driven vehicles in a mixed traffic environment. Lane change behavior, especially for mandatory lane change, is a typical example of personalized driver behavior that needs to be modeled and be predicted. Lane change is one of the trickiest since it requires the tacit cooperation of lateral control and

longitudinal control from the driver. Therefore, compared with the prediction of longitudinal maneuvers such as car-following, which is heavily correlated with the gap between the ego vehicle and the leading vehicle, the prediction of lane change is much more complicated and challenging. The online lane-change prediction of human drivers becomes essential for IV since it provides inputs to the downstream motion planners and controllers and hence allows IV to better cooperate with surrounding human-driven vehicles. By building the personalized lane change model, CAVs can have a better understanding of the specific human drivers and provide a more accurate prediction.

This section presents a learning-based algorithm for online lane change prediction utilizing personalized behavior modeling. Field implementations are carried out on a customized vehicle-edge-cloud platform under the digital twin framework. In the field implementation, the cloud (i.e., Amazon Web Services) analyzes the personalized behavior of human-driven vehicles with connectivity (e.g., by cellphone) and stores the learned driver models and historical data, while the edge (i.e., local server) is responsible for the computation of online lane change prediction.

Compared to the existing literature on prediction and behavior modeling, this study has the following main contributions:

- A hierarchical learning-based system is developed for personalized driving behavior modeling, online lane change prediction, and trajectory likelihood estimation.
- Under the digital twin framework, a vehicle-edge-cloud platform is constructed and demonstrated, enabling real-world data collection and algorithms development.

- Based on density-based spatial clustering of applications with noise (DBSCAN), we develop an unsupervised data labeling method by adding temporal information to relate adjacent data points in a time series.
- To validate the proposed algorithm in the field experiments, a portable vision-based human-machine interface (HMI) system is designed to provide prediction information to the driver supported by edge computing and cloud computing.

4.3.1.2 Specifications and Assumptions

In this section, the target predicted vehicle is a connected human-driven vehicle, whose historical/real-time data and the trained driver model are accessible through the digital twin. When other connected vehicles detect and recognize this target vehicle, they can download the driver model of the target vehicle to assist in the prediction. Specifically, our prediction algorithm is designed for the on/off-ramp scenario to predict the maneuver and trajectory of the on-ramp vehicle.

To expedite field implementation, some reasonable specifications and assumptions are made in the current stage of online lane change prediction and personalized driving behavior modeling, as follows:

- When the target vehicle comes into view, if the ego vehicle is able to recognize it (e.g., by computer vision or V2X communications), the associated driver model is then acquired from the cloud server (with permission). Before the target vehicle is recognized, a general driver model will be used instead.
- The driving preference of the same driver is assumed to be long-lasting and will not be affected by the mood on the testing day.

- As designing the perception system (e.g., LiDAR, radar, and camera) is outside the scope of this dissertation, the necessary vehicle information (e.g., location and yaw angle) is uploaded by the target vehicle and shared by the edge server.

4.3.1.3 Methodologies

The proposed system for lane-change behavior prediction is shown in Fig. 4-23. The system consists of an offline learning process and an online validation process. In the offline learning process, based on the dataset collected on the Vehicle-Edge-Cloud digital twin platform, a personalized Seq2Seq LSTM model is trained to predict the lane-change decision, and the cost function (as discussed in Section 4.2.3) inferring the driver preference is learned by the Inverse Reinforcement Learning (IRL).

In the online validation process, at each time step, the vehicle states will be analyzed by the Seq2Seq LSTM network to recognize the maneuver and select a proper cost function. Next, the cost function evaluates the confidence of possible trajectories provided by the trajectories generator. Finally, the outputs including the most probable trajectory and lane change probability are visualized and sent back to the edge server, and the major results of each phase are highlighted with red boundary.

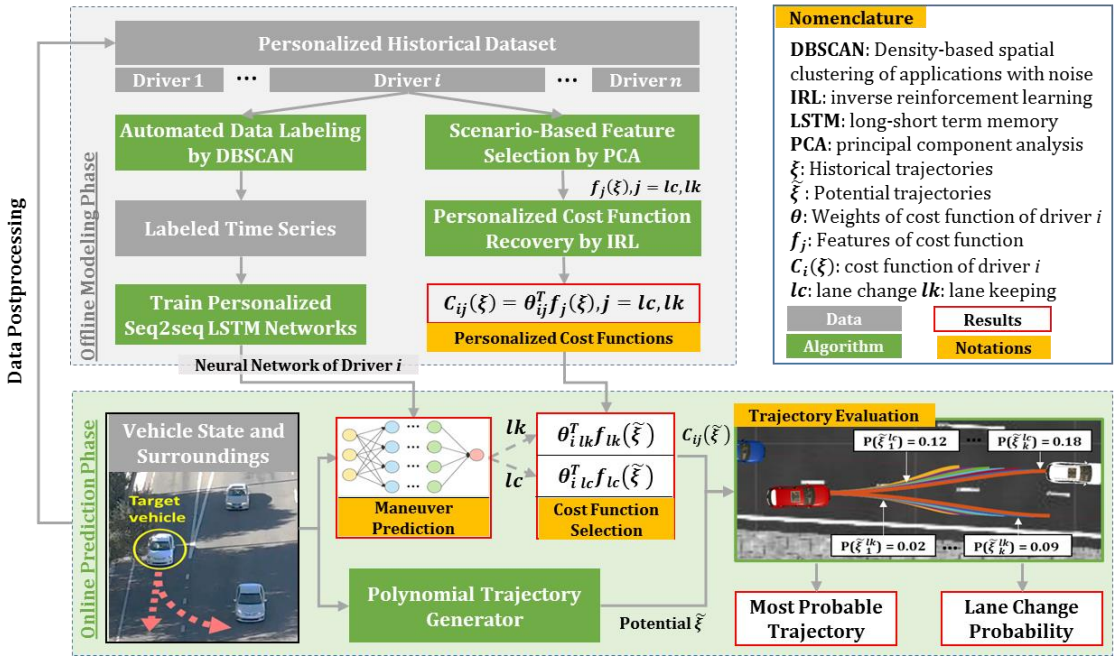


Fig. 4-23 Personalized lane change behavior modeling: offline learning and online prediction

4.3.1.3.1 Lane-change maneuver prediction

In order to analyze the trajectory in detail, we need to recognize the driver's intention. We assume that before planning the trajectory, the human driver first considers high-level tasks (e.g., lane change and lane keeping). Therefore, the lane-change intention prediction is formulated as a time-series classification problem, which predicts vehicle states in future time steps, i.e., either lane change or lane keeping. That is to classify the future T-step actions $A_{t:t+T}$ into $\{a_{\text{change}}, a_{\text{keep}}\}$, given historical vehicle states and the map information.

To model long-term temporal dependencies among time series, LSTM network is chosen, as its time series prediction capability is validated in many existing studies, e.g., [138]–[140]. Since each vehicle state in the time series is highly correlated with its adjacent time steps, the Seq2seq neural network using two LSTMs [141] is adopted for a multi-step and multivariable prediction.

The prediction model for a driver is built based on the historical trajectories $\Xi = \{\xi_k\}, k = 1, \dots, K$, and a trajectory contains vehicle states at every time step, i.e., $\xi = (s_1, \dots, s_t)$, where s_t is a vector and denotes the vehicle state at t timestep. The vehicle states consist of the information of ego vehicle and its surrounding environment, which can reflect the operation and the perception of the driver.

The structure of our network is shown in Fig. 4-24. The neural network input is a trajectory sequence $\xi = \left(s_{t-T+1}^{lstm}, \dots, s_t^{lstm}\right)$ of the last T steps, and the vehicle states s_t consists of yaw angle, lateral speed, longitudinal speed, and remaining distance for a mandatory lane change. The output is the predicted lane-change action sequence $(A_{t+1}, \dots, A_{t+T+1})$ for the next T steps. The network consists of two LSTM layers (each followed by one dropout layer) and two fully connected layers (with ReLu and Softmax layer as their activation layers). The labeled dataset is split into the training set, validation set, and test set. Since the units of vehicle state features are different, the whole dataset is normalized to the range of [0,1].

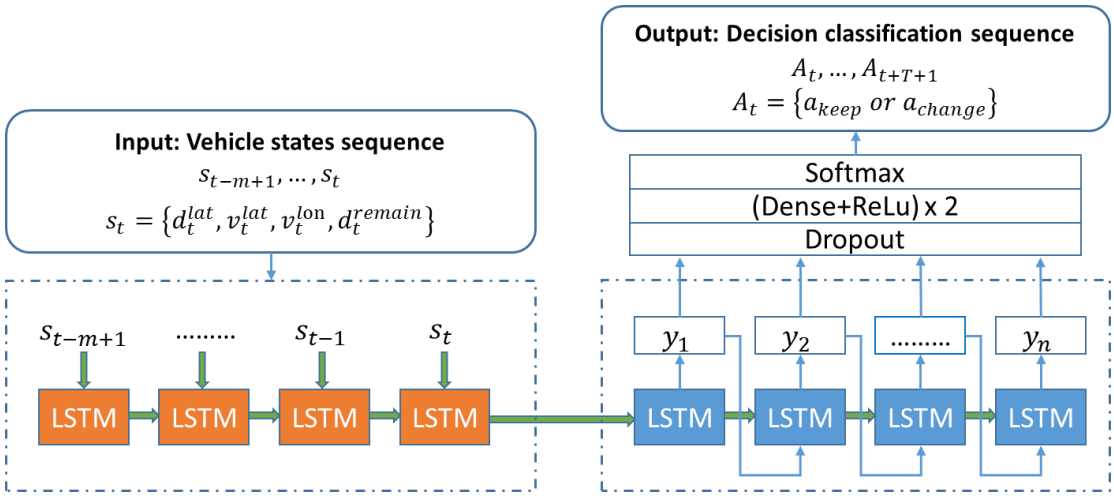


Fig. 4-24 Structure of the proposed neural network for lane-change decision prediction

4.3.1.3.2 Unsupervised data labeling based on modified DBSCAN

To predict the lane-change maneuver, we need to recognize those lane-change moments among the dataset, so labeling each time step for the dataset is the first step of data processing. The raw trajectory data is integrated with the map information to create a dataset that contains features potentially affecting driver's lane-change decision making, and these factors include vehicle speed, position, distance to surrounding vehicles, speed of surrounding vehicles, deviation from the lane centerline, remaining distance to the mandatory lane-change point (if any), and speed limit.

We separate the trajectories that contain lane-change maneuvers from the whole dataset, by monitoring the accumulated lane deviation. Then those trajectories are further processed for recognizing the decision of the driver at each time step. Inspired by [142], we applied DBSCAN to label lane-change and lane-keep maneuvers for each vehicle state at each time step. The lateral speed (v_{lat}) and lateral acceleration (a_{lat}) are used as two input features for DBSCAN, and the outputs are two clusters of vehicle states, e.g., lane change or lane keep.

However, DBSCAN does not consider temporal relation among the data points and hence cannot guarantee the continuity of the lane-change maneuver. To eliminate the noise of the labeled time series, a morphological operation [143] is applied to the dataset after DBSCAN clustering, as described in Algorithm 3. For example, in this study, we apply $M_t = [1 \quad ... \quad 1]_{1\times 5}$, as we assume the lane-change maneuver is continuous in a short period and at least lasts for 0.5 seconds with an update rate of 10 Hz.

Algorithm 3 Temporal Filter: denoise the labeled time series based on the temporal characteristic of lane-changing maneuver.

Input: 1. Labeled time series data T by DBSCAN. 2. A morphological structuring element (M_t) for the temporal characteristic of lane changing.

Output: Continuous denoised time series.

-Morphological closing operation-

1: Calculate the dilation ($\oplus$) of T by M_t: $T_1 = T \oplus M_t = \{z \in E \mid (M_t{}^s)_z \cap T \neq \emptyset\}$, where E is a Euclidean space or an integer grid, $M_t{}^s = \{x \in E \mid -x \in M_t\}$, and $(M_t{}^s)_z$ is the translation of $M_t{}^s$ by the vector z, i.e.,$(M_t{}^s)_z = \{b + z \mid b \in M_t\}, \forall z \in E$;

2: Calculate the erosion ($\ominus$) of T_1 by M_t:$T_2 = T_1 \ominus M_t = \{z \in E \mid M_{t_z} \subseteq T_1\}$, where M_{t_z} is the translation of M_t by the vector z;

-Morphological opening operation-

3: Calculate the erosion of T_2 by M_t: $T_3 = T_2 \ominus M_t$;

4: Calculate the dilation of T_3 by M_t: $T_{ult} = T_3 \oplus M_t$;

4.3.1.3.3 Trajectory evaluation

To execute the decision of lane change or lane keeping, planning the trajectory is essential. Considering the real-time performance, instead of exploring arbitrary trajectory, we adopt a polynomial trajectory generator [45] to plan the candidate trajectories $\tilde{\xi}_k$, As in Fig. 4-25, at each time step, this trajectory generator takes the vehicle's state $\{x, y, v, a, yaw\}$, as inputs and generates multiple trajectories within a prediction window. With integrated vehicle kinematics, the trajectory generation ensures that the prediction result is realistic and reachable. In this study, we set the planning time window as 4 seconds.

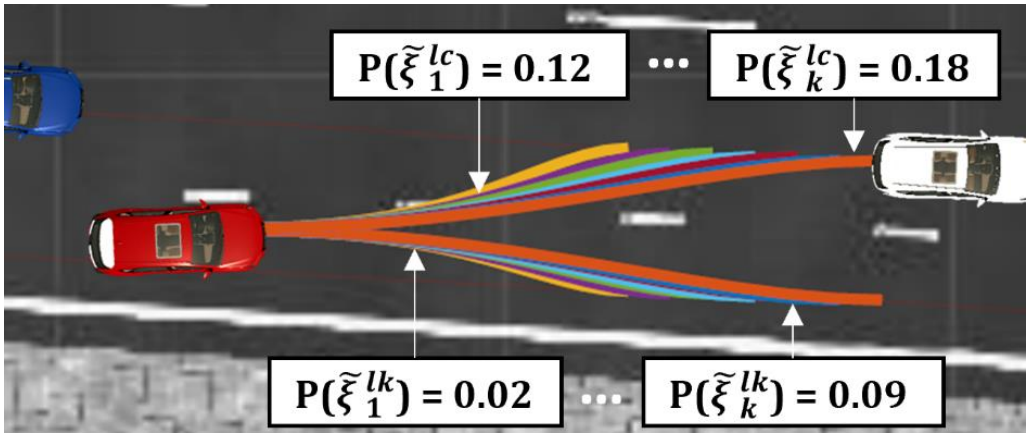


Fig. 4-25 Polynomial trajectory generator

Based on Equation 25, the cost function $C_i(\theta_i, \tilde{\xi}_k)$ is used to evaluate the probability of each possible trajectory $\tilde{\xi}_k$, and select the most probable trajectory. The probability of the lane change maneuver prediction is evaluated by Equation 26, i.e., the probability of lane change equals the sum of the probabilities of all sampled lane-change trajectories.

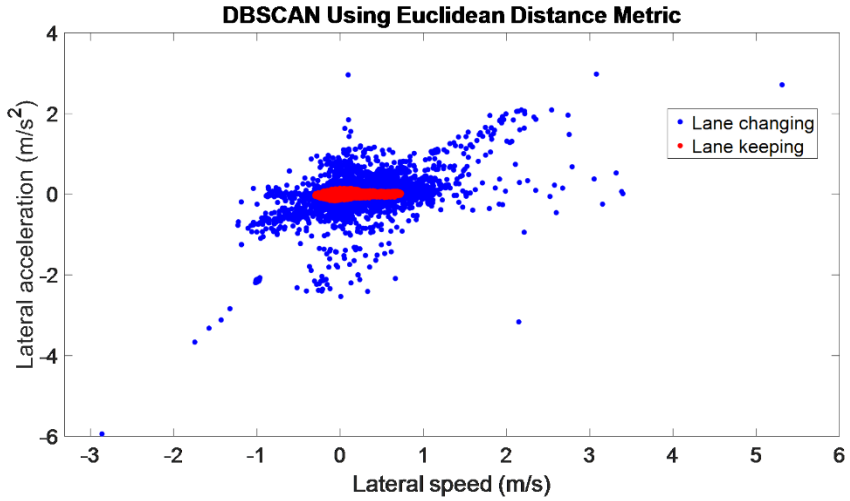
$$P(\xi_k \mid \theta_i^*) = \frac{e^{-C_i(\theta_i^*, \xi_k)}}{\sum_{k=1}^{K} e^{-C_i(\theta_i^*, \xi_k)}} \tag{25}$$

$$P(\hat{a}_i) = \sum_{k=1}^{K} P(\tilde{\xi}_k \mid \theta_i^*) \tag{26}$$

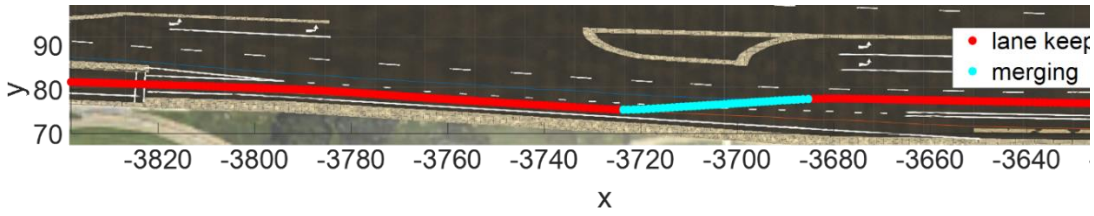
4.3.1.3.4 Algorithm validation in simulation

Before modeling the personalized driving behavior in the real world, the prediction capability of the proposed algorithm is validated on a general driving behavior model in simulation. In the human-in-the-loop co-simulation (as discussed in Section 4.1.2) platform, a real-world test track is programmed in the Unity game engine as the digital version of the real-world on/off ramp area in the Vehicle-Edge-Cloud digital twin testbed (as discussed in Section 4.1.1). In the simulation, a mixed traffic flow is generated by SUMO, and human input is consumed by Unity with Logitech driving set, allowing various

drivers to conduct human-in-the-loop simulations in an immersive traffic environment, where drivers can experience mixed traffic with different CAV penetration rate and congestion levels. To model the general lane-changing and lane-keeping behavior, 59 trips are collected from ramp drivers within the on-ramp/off-ramp area, under a volume-to-capacity (V/C) ratio of 0.6.



(a) Clustering result



(b) Data labeling for each time step

Fig. 4-26 Trajectory labeling based on DBSCAN

To be specific, in this study, 37 trips with lane changes and 22 trips without any lane change within the on-ramp/off-ramp area are collected. The average duration of each trip is 30 seconds, with an update rate of 10 Hz. This data is processed by DBSCAN as shown in

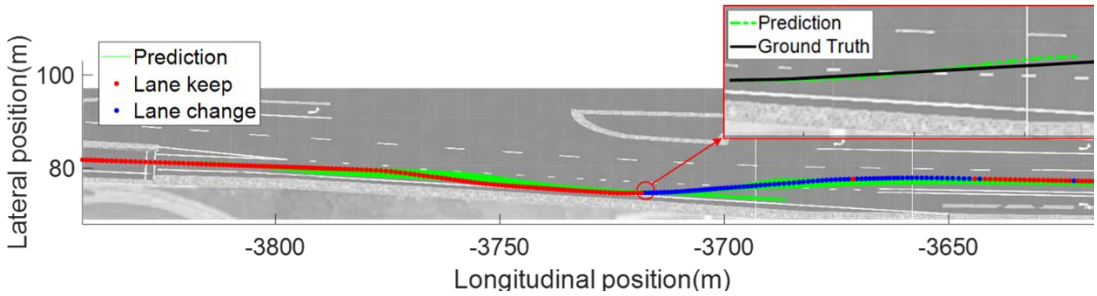
Fig. 4-26 (a), and an example of labeled trajectory is shown in Fig. 4-26 (b), indicating that the lane-change segment is well labeled.

The real-time predicted lane change probabilities and trajectories are visualized in Fig. 4-27, and the proposed algorithm recognizes the lane change maneuver in 3 seconds before the vehicle crosses the lane separation line. At each time step, the current vehicle state is sent into the LSTM network for decision prediction, where look-backward and prediction windows are both 3 seconds (30-time steps). The decision prediction result guides the system to the corresponding cost function which will be used to evaluate all the trajectory candidates. Based on Equation 25, we select the most probable trajectory as our prediction, and it can be projected into the simulation platform, where the probability of lane change is also estimated based on Equation 26.

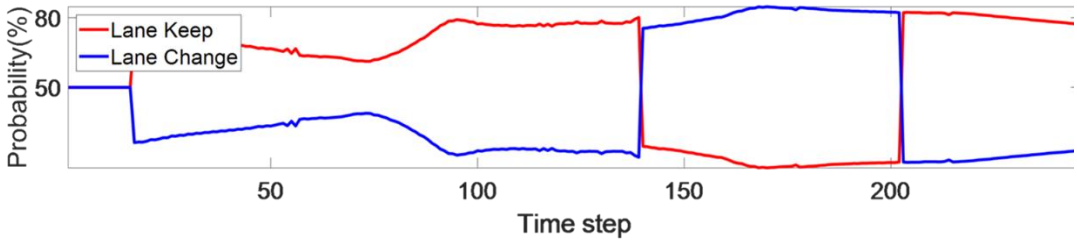
Fig. 4-27 Algorithm validation on UCR's human-in-the-loop co-simulation platform: predictions for lane change and most probable trajectory.

Fig. 4-28(a) presents the whole prediction process of the same trip of Fig. 4-27. Each time step of the ground-truth trajectory is labeled by LSTM in real time as lane keep (red dots) or lane change (blue dots). This prediction result shows that the lane-change decision

is recognized in 3 seconds (30-time steps) before the vehicle crosses the borderline. Also, the visualized comparison of the 4-second horizon predicted trajectory (in green dash line) with the ground truth is shown in the zoom-in subfigure. Fig. 4-28(b) depicts the probability estimation of lane change and lane keep during a trip containing a lane-change maneuver, reflecting the intention of the driver. In addition, Fig. 4-29 displays a trip without any lane-change behavior, and the confidence of the prediction increases as the vehicle gets closer to the end of the lane-change area.

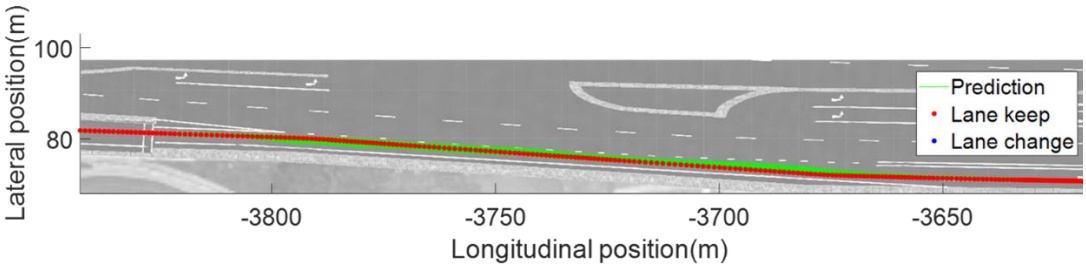


(a) The whole process of lane-change prediction

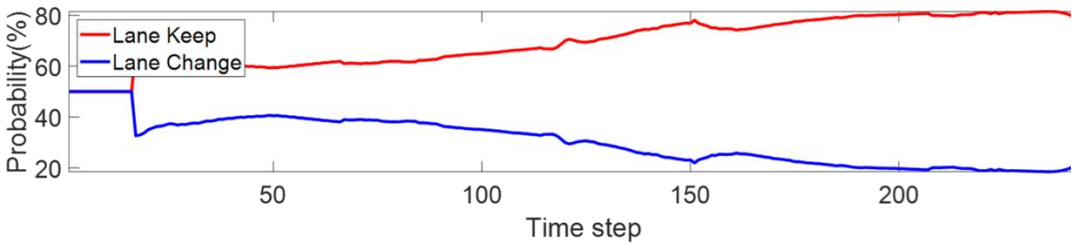


(b) Predicted Lane change probability over time

Fig. 4-28 Online lane-change prediction of a trip with one lane change event



(a) The whole process of lane-change prediction



(b) Predicted Lane change probability over time

Fig. 4-29 Online lane-change prediction of a trip without any lane change event

Moreover, the Mean Euclidean Distance (MED) [144] is used to quantify the accuracy of trajectory prediction. At time step t, the predicted trajectory$\hat{\xi}_t(L) = \{\widehat{xy}_t, \dots, \widehat{xy}_{t+L}\}$ is compared with the ground truth $\xi_t(L) = \{xy_t, \dots, xy_{t+L}\}$ within the same horizon L and the same sampling rate, as shown in Equation 27, where $xy = (x_t, y_t)$.

$$m_{\mathrm{MED}}(\hat{\xi}_t, \xi_t) = \frac{1}{L}\sum_{l=1}^{L} \|\widehat{xy}_{t+l} - xy_{t+l}\|_2 \tag{27}$$

In our online test, this general model achieves a mean MED of 0.39 m on average within a 4-second prediction window for 10 test trips, outperforming the IRL-based prediction method in [13], which achieves a mean MED of 0.62 m in a 3-second prediction window. More importantly, we validate our methodology in an online fashion that allows drivers to test the system in human-in-the-loop simulations, while most other literature only did this in an offline fashion by running numerical simulations. In the field implementation, we further study how the personalized model improves prediction.

4.3.1.4 Description of Field Operational System

The field implementation is carried out on the vehicle-edge-cloud digital twin platform, as introduced in Fig. 4-1, while its hardware is explained in Fig. 4-2. The real-world implementation is conducted by three passenger vehicles (as in the vehicle level of the vehicle-edge-cloud platform), which are mainline vehicle 1 (MV1), mainline vehicle 2 (MV2), and ramp vehicle (RV), respectively. Specifically, RV is the target predicted ramp vehicle, MV1 is the conflicting vehicle with RV. While MV2 drives behind MV1 and is potentially affected by the lane change maneuver of RV, MV2 observes and predicts the whole lane change process, as described in Fig. 4-30. To ensure MV1 and RV can encounter each other, speed guidance is provided to MV1 only before reaching an observable point, from which MV1 and RV can see each other for the rest of the trip. The speed guidance [87] is calculated by edge server, where RV has its virtual projection on the mainline, and MV1 is assigned to follow the virtual projection without an intervehicle gap. By following the speed guidance, MV1 can reach the observable point at the same time as RV, and for each experiment, nearly the same condition of merging conflict is guaranteed. Unlike MV1, to collect naturalistic driving data of the RV driver, RV receives no interference from the system during the whole trip. During the conflict creation process, MV2 is not involved since it plays the role of an observer and has no impact on the lane change. If MV2 successfully detects and recognizes RV, MV2 sends a request and downloads the driver behavior model of RV from the cloud to facilitate the lane change prediction.

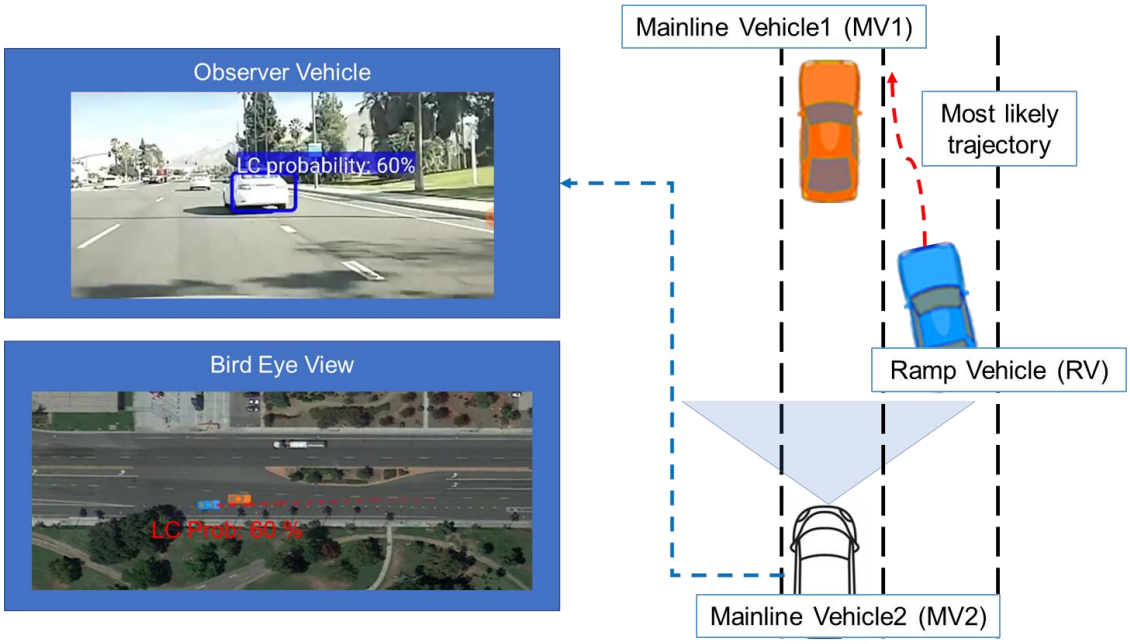
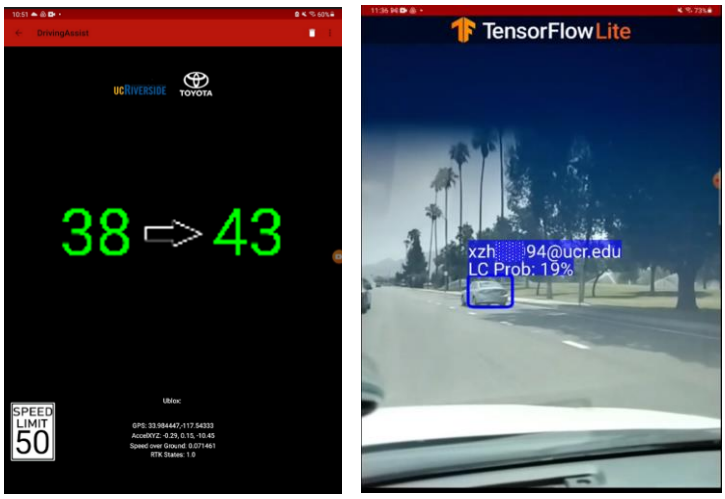


Fig. 4-30 The field implementation plan for lane change prediction.

To meet the above demands, the HMI guides MV1 with speed guidance which shows the current speed on the left and the target speed on the right as shown in Fig. *4-31* (a). For MV2, we leverage the benefits from TensorFlow Lite Object Detection, an open-source Android application, to detect the RV and visualize the predicted lane-changing probability with a user ID on the top of a bounding box as indicated in Fig. *4-31* (b).



(a) (b)

Fig. 4-31 Vehicle HMI design. (a) Speed guidance, and (b) detection for lane changing prediction

Before starting the vehicle, each driver logs in with a unique ID and password to access his/her personalized digital twin. In addition, to obtain personalized services, drivers need to agree with sharing information with edge server.

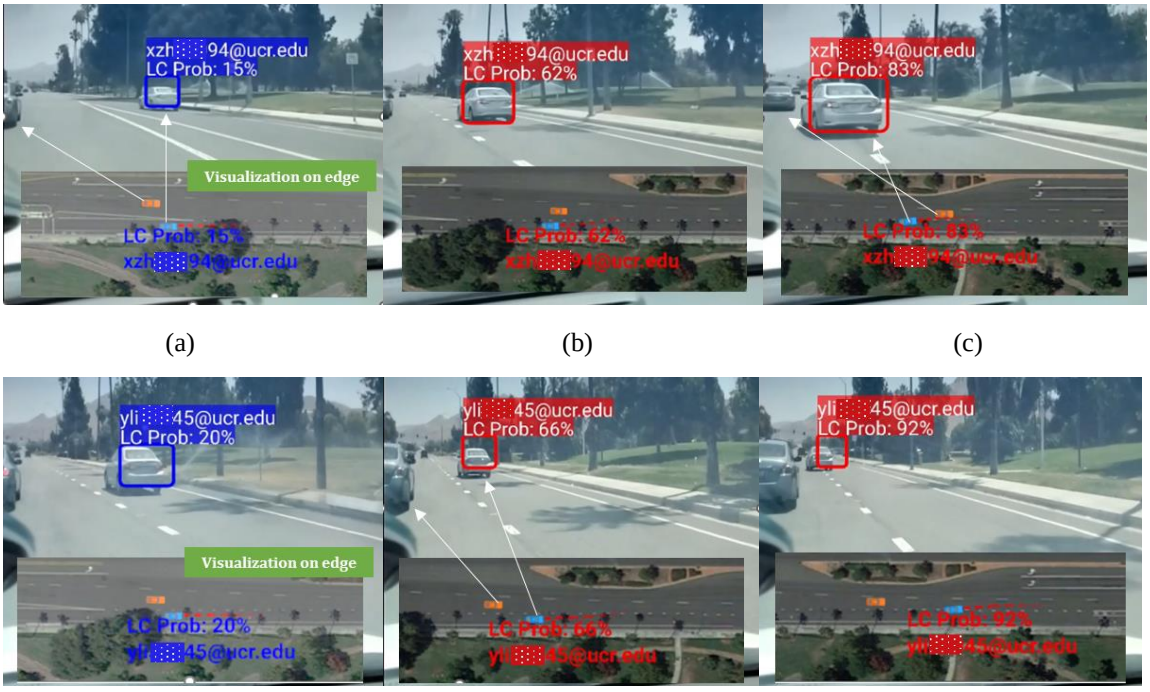
A bird-eye view Android application is designed for visualizing the vehicle Digital Twins running on the edge server. MV1 and RV are the orange vehicles and the blue vehicle respectively in Fig. *4-32*. Moreover, the lane changing probability and the predicted trajectory of RV are provided for a better understanding of the entire field implementation process.

Fig. 4-32 The bird-eye's view application interface: digital twin of vehicles running on the edge server in real time.

4.3.1.5 Result Analysis

As presented in Fig. 4-33, the online prediction combines probability estimation and trajectory prediction. In stage 1 see Fig. 4-33(a), RV enters the interacting zone and is recognized by MV2 (observer). At the same time, MV2 sends a request to edge server, and then edge server receives the driver model from the cloud. Then, the driver ID and lane change probability are visualized on both observer's view and edge. Meanwhile, the lane change probability shown above the blue bounding box is low, and the predicted trajectory (visualized on edge in red dash line) is straight, indicating the lane-keeping maneuver. In stage 2 see Fig. 4-33(b), RV slows down to yield MV1, which is captured by MV2. At the moment, the predicted trajectory points to the lane separation line, and the lane change probability increases with the color of the bounding box changing from blue to red for warning of a potential lane change. In stage 3 see Fig. 4-33(c), Driver 1 perceives that the gap is large enough, and the lane change begins. MV2 is more certain about the prediction, showing a high probability and a predicted trajectory pointing to the center of the mainline. Fig. 4-33(d)(e)(f) present a similar process for Driver 2's lane change. One noticeable

difference from Driver 1's model is that Driver 2's personalized model pays attention to speeding up rather than slowing down.



(a) (b) (c)

(d) (e) (f)

Fig. 4-33 Field implementation for online lane change prediction with visualization on camera and edge server: (a)(b)(c) Prediction for Driver 1, changing the lane behind MV1. (d)(e)(f) Prediction for Driver 2, changing the lane in front of MV1.

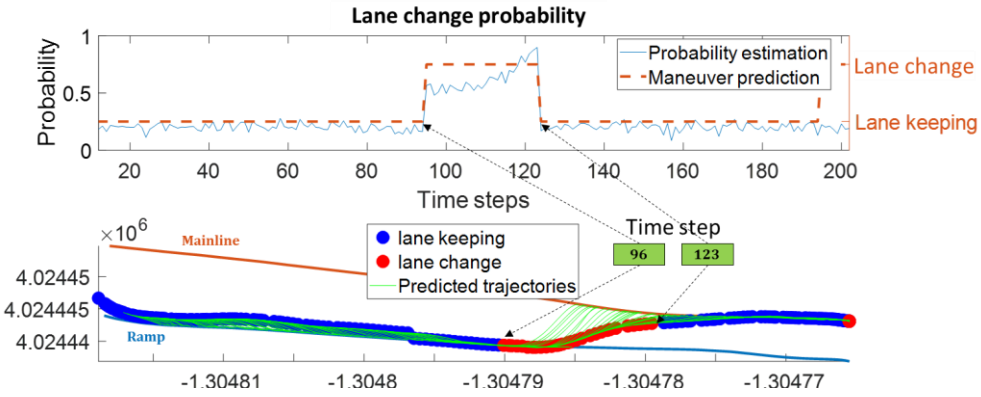
Fig. 4-34 elaborates on the whole prediction process of the same trips in Fig. 4-33. Specifically, Fig. 4-34 (a) presents the prediction result of one lane change performed by Driver 1. The top subfigure depicts the prediction result and probability estimation of lane change during a trip, reflecting the intention of the driver. In the bottom subfigure, each time step of the ground-truth trajectory is labeled by the neural network in real time as lane change (red dots) or lane keeping (blue dots), when the predicted trajectory of each time step is shown in the green line.

The maneuver prediction can be corrected by probability estimation. In the top subfigure of Fig. 4-34 (b), when the lane change has been completed, lane change is still predicted by LSTM (orange dash line) for the time steps from 177 to 180, but the probability of lane change (blue solid line) is estimated only 20%. According to the bottom subfigure, Driver 2 has completed the lane change at the 177th step, and the cost function-based probability estimation corrects the neural network prediction.

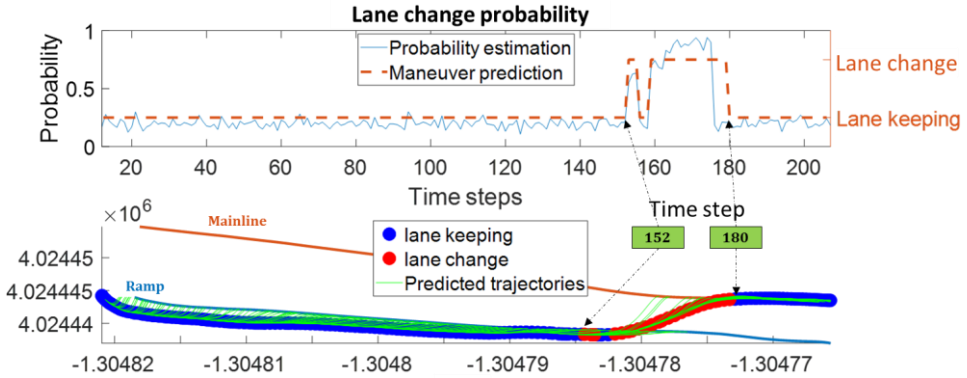
Personalized models can only be used on a specific person. Fig. 4-34 (c) illustrates how will the prediction result be when the model is mismatched (using Driver 1's model on Driver 2's trip), where the predictions are not accurate for lane change probability, action, and trajectory.

The prediction needs to be evaluated for both maneuver and trajectory. For lane change intention recognition, the predictive capability can be quantified as the time between the moment of recognizing the lane change intention and the moment of the vehicle crossing the lane separation line. The personalized model of Driver 1 recognizes the lane change intention in 6.08 seconds on average, with a 1.96-second standard deviation (STD), while Driver 2's model achieves an average of 3.73 seconds with an STD of 1.29 seconds. In this study, output of intention recognition layer can be corrected and modified by the downstream probability estimation. Therefore, we propose a heuristic method to evaluate accuracy of intention recognition based on the lane change probability. It should be noted that the accuracy can be evaluated in a more sophisticated way. If the probability of the moment when the vehicle touches the lane separation line (P_s) is significantly larger than the probability during lane keeping (P_k), i.e., $\sigma \cdot P_s > P_k$, the lane change is successfully

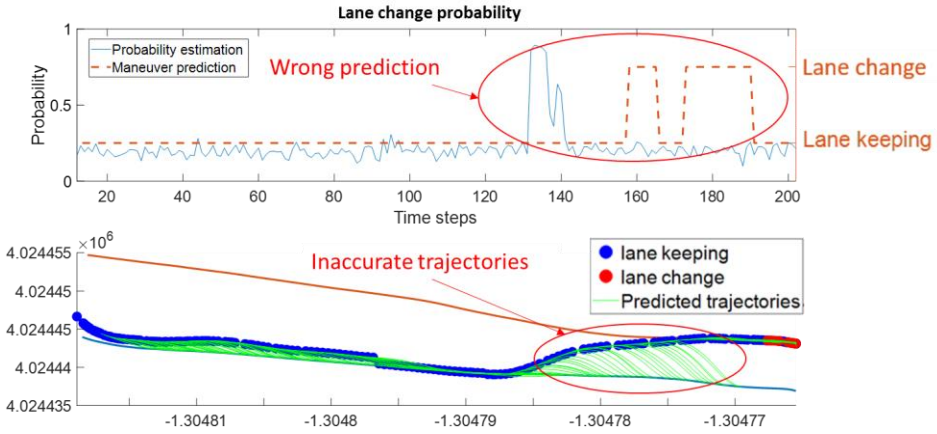
predicted. The significant factor σ can be chosen as 0.5. The proposed general model predicts 90.4% lane change events, while the personalized model for Driver 1 exhibits an even higher level of performance with a prediction accuracy of 95.2%.



(a) Online prediction process for Driver 1



(b) Online prediction process for Driver 2



(c) Offline prediction using a mismatched driver model

Fig. 4-34 Prediction result analysis of using personalized models

For trajectory prediction, the Mean Euclidean Distance (MED) is measured to analyze the accuracy. Although we cannot decouple the GPS error and prediction error, the model

achieves a good result. The quantitative evaluation is shown in Table VIII, which presents the MED of baseline, general model, and personalized model. Inspired by [59], we adopt Seq2seq LSTM as the baseline model for the entire pipeline, while the proposed method employs the Seq2seq LSTM solely for the intention recognition part in the overall pipeline. Within a 4-second prediction window, the general model of the proposed algorithm outperforms the baseline.

Using either a general or personalized model, the prediction of Driver 1 is better than Driver 2. The MEDs between the predicted trajectory and GPS points are 1.03 m (STD: 0.4 m) for Driver 1 and 1.48 m (STD: 1.05 m) for Driver 2 in a 4-second prediction window. Compared to results from the general model, the personalized model improves Driver 2's results the most, by 27.8% on average. Since Driver 1 is more predictable, the improvement of using a personalized model is limited (by 1.9%), but the prediction variation is reduced by 42%.

TABLE VIII PREDICTION ACCURACY COMPARISON IN A 4-SECOND PREDICTION WINDOW

MED (m)		Seq2seq LSTM	General	Personalized	Improvement in the general model
Driver 1	Mean	6.76	1.05	1.03	1.9%
	STD	1.16	0.69	0.4	42%
Driver 2	Mean	8.28	2.05	1.48	27.8%
	STD	1.99	1.17	1.05	10%

4.3.1.6 Conclusions and Future Work

This dissertation has proposed an online lane change prediction algorithm based on personalized driving behavior modeling, which is validated on a vehicle-edge-cloud

testbed under the Driver Digital Twin (DDT) concept. Specifically, a sequence-to-sequence LSTM neural network has been used to predict the lane change intention, and personalized driving behaviors have been modeled for different drivers, whose preferences are learned and analyzed by inverse reinforcement learning based on the historical data stored on the cloud server. Supported by personalized models, an online lane change prediction system has been developed and validated with real-world field implementation. The system is able to recognize the target driver's lane change intention at 6.08 seconds before the vehicle crosses the lane separation line, and the Mean Euclidean Distance between the predicted trajectory and ground truth (based on the measurements from an RTK-enabled GPS unit) is 1.03 m within a 4-second prediction window. Using a personalized model can improve the prediction accuracy by 27.8%.

As one of the first few research projects looking into personalized driving behavior, there are still some limitations of this implementation, and improvements can be made alongside the future development. The prediction algorithm is specifically designed for on/off-ramp mandatory lane changing. The mechanism behind discretionary lane changing maneuvers can be different, which requires adjustment on feature selection of the cost function. Another major constraint on studying personalized behavior is data availability, which can be relieved by transferring the model learned from simulation. Besides the research on personalized behavior modeling, incorporating personalized prediction with planning is also an important future step, as it allows CAVs to drive like human-driven vehicles, and improve their user acceptance and trust.

4.3.2 Ramp Merging Strategy with Feedforward Planning and Feedback Control

4.3.2.1 Problem Statement

Building upon the previous research on lane change prediction based on personalized behavior modeling, we seek to expand our understanding and analysis to encompass the complex dynamics of on-ramp merging. By studying the intention and trajectory of vehicles during on-ramp merging, we can gain valuable insights into the decision-making processes of drivers, their interaction with surrounding traffic, and potential risk factors involved. In this section, we expand the behavior study to traffic level, and explore an agent-based planning algorithm to coordinate the on-ramp mering.

Near the on-ramp merging area, each vehicle needs to identify whether the merging conflict exists and adjust the speed to ensure a smooth and safe merging process while maintaining the overall flow of traffic. After predicting the merging behavior of surrounding vehicles and knowing the conflict cannot be avoided, a game between a mainline vehicle and a ramp vehicle is played in each time step whenever a conflict exists [145]. The game starts when the conflict emerges and ends until this conflict is solved. During the merging process, complex conflict can be summarized with three types of scenarios, including the interactions between 1) two legacy vehicles, 2) two CAVs, and 3) a CAV and a legacy vehicle. This chapter will only discuss the CAV(s) involved conflicts since the conflicts between two legacy vehicles cannot be coordinated directly by CAVs. Hereafter, this section will analyze the merging strategy from the perspective of the ego vehicle (CAV).

4.3.2.2 Introduction and Background

Traffic-related issues such as safety, efficiency, and environmental sustainability have drawn significant attention as transportation is more involved in people's daily lives. Among the factors leading to traffic congestion and accidents, ramp merging has a significant amount of impact [146]. Vehicles merging near the ramp area have been a major concern that generates numerous potential conflicts, due to the chaotic nature of driving behaviors and the lack of coordination in the merging area. The difficulty arises for drivers of ramp vehicles along the on-ramp, where drivers must discern to accelerate/decelerate to enter the mainline safely without a clear line of sight regarding the mainline traffic. Meanwhile, drivers of mainline vehicles may have to modify their vehicle speeds to permit the entrance of ramp vehicles, thus affecting upstream traffic flows and consuming excessive energy.

The emergence of connected and automated vehicle (CAV) technology brings about solutions to ramp merging issues. By taking advantage of vehicle-to-everything (V2X) communications, vehicles can communicate with other road participants. As a typical example of CAV technology implementation, the cooperative merging of vehicles at ramp has been studied and applied by various researchers around the globe, where connected vehicles communicate with vehicles coming from the other lane directly or through roadside infrastructure, and hence conduct cooperative merging maneuvers in a safe and smoothed manner [147]–[149].

Since CAVs are supposed to share the road with legacy vehicles in the short future, considering the mixed traffic environment is more pragmatic, though more challenging in

terms of regulating the entire traffic stream. The well-planned operation for CAVs may be interrupted by legacy vehicles, hence the interaction between CAVs and legacy vehicles should not be ignored. Specifically, CAVs need to understand human-driven vehicles' behaviors, make decisions dynamically regarding the actions to be taken, and execute such actions through the planner and controller. Therefore, many researchers incorporate driving behavior modeling into their planning and control design [11], [150], [151], recognized as the driver behavior-aware system.

In this section, a decentralized feedforward/feedback planning algorithm based on game theory is proposed to coordinate the on-ramp merging, providing the optimal merging sequence and associated speed trajectory for each CAV in the mixed traffic. Compared to other recent studies on cooperative ramp merging, the major contributions of this study are as follows:

- For the mixed traffic scenario, a synthetic agent-based ramp merging strategy with both lateral and longitudinal control is proposed, including functions of conflict prediction, conflict avoidance, merging sequence determination, acceleration, and lane change control.
- A game theory-based ramp merging sequence decision-making method for CAVs is developed by considering different costs, such as safety, mobility, and comfort. Moreover, cooperative and non-cooperative games are formed for different types of interaction.
- A traffic flow level simulation is carried out on a uniquely developed Unity-SUMO co-simulation platform to validate the algorithm and analyze the result.

4.3.2.3 Methodology

This section presents the proposed ramp merging strategy for CAVs in mixed traffic. The system architecture and strategy workflow are introduced in the first subsection, followed by the elaboration of the game theory algorithm and decision-making process.

4.3.2.3.1 Strategy workflow

Our strategy is designed from a decentralized agent-based model perspective, allowing vehicles to act independently. The strategy workflow is shown in Fig. 4-35, where every vehicle goes through this process at each time step. Six major modules are functioning to support the strategy.

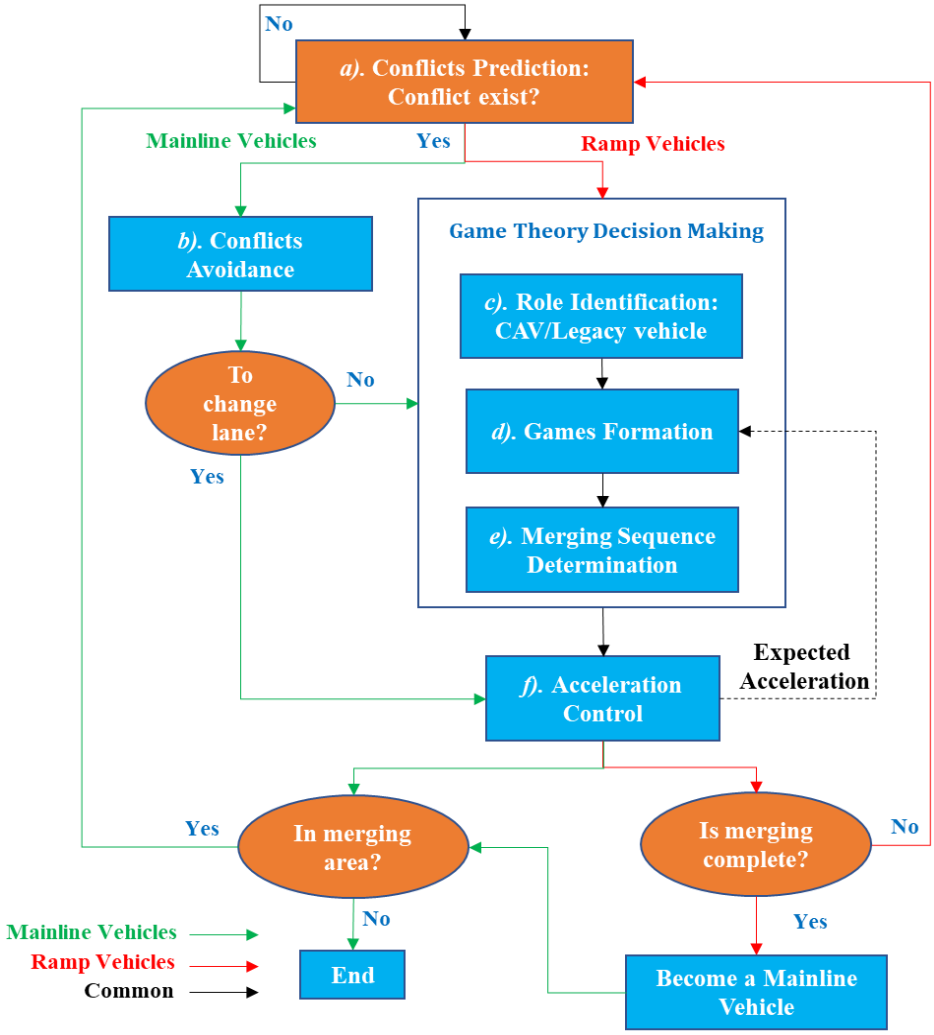


Fig. 4-35 System workflow of the mixed traffic ramp merging strategy for CAVs

a) Conflict Prediction Module: Based on the information from the radar system or other CAVs, this module projects the ego vehicle and its surrounding vehicles into the future to see whether conflicts exist in the next time step. As Fig. *4-36* illustrates, when the projected surrounding vehicle V_j' is out of the safe distance of the projected ego vehicle V_i', there is no conflict; otherwise, this surrounding vehicle will be classified as a potential conflict and added to the conflict list. The analytical form of projection and conflict prediction can be expressed in Equation 28. As the conflict state can be an instant condition and be too

sensitive, a hysteresis controller considering historical information is implemented to filter the conflict state results [152].

$$No\ conflict, \quad \begin{cases} \hat{v}_i \times \Delta t + D_{safe} - d_{ij} \le \hat{v}_j \times \Delta t \\ or \\ \hat{v}_i \times \Delta t - D_{safe} - d_{ij} \ge \hat{v}_j \times \Delta t \end{cases} \tag{28}$$

$$Potential\ conflict, \quad else$$

where $\hat{v}_i$ is the predicted speed of ego vehicle; $\hat{v}_j$ is the predicted speed of its surrounding vehicle, and these predicted speeds are from the lane change prediction algorithm discussed in Section 4.3.1. $d_{ij} = x_i - x_j$ is the current clearance; Δt is the simulation time step length; D_{safe} is the safe clearance, which is a speed-variant term depending on both the minimum static clearance and safe time headway of the ego vehicle.

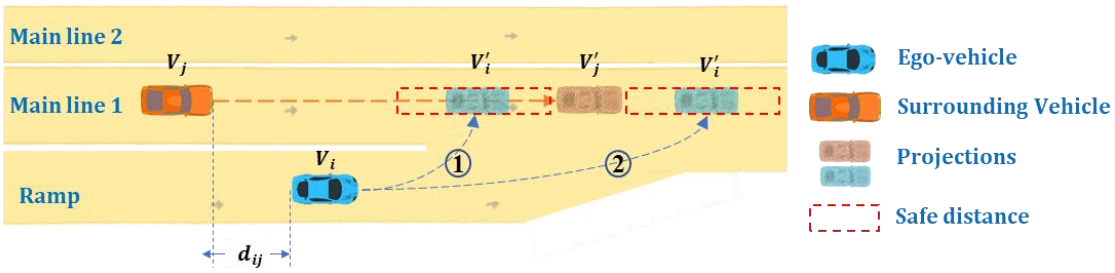


Fig. 4-36 Conflict prediction module

b) Conflict Avoidance Module: Avoiding conflict is the preferred option taken by mainline vehicles, who can change to another lane to avoid conflict with ramp vehicles. This module takes time-to-collision (TTC) and inter-vehicle gap into account, urging mainline vehicles to change lanes for larger inter-vehicle gaps and TTCs. When mainline vehicles cannot avoid conflict by changing lanes, mainline vehicle activates the *Game Formation Module* and initiates the game for determining the merging sequence with ramp vehicles.

c) Role Identification Module: This module is developed to classify ego vehicle's potential competitors into either CAV or legacy vehicle, which determines the game type in *Module d)*. Basically, the vehicle type (CAV or legacy vehicle) can be identified based on the signal association from both the radar detection and wireless communication.

d) Game Formation Module: The game formulation process is triggered once it receives a conflict notice from *Conflict* Prediction *Module,* where ego vehicle forms an individual (two-player) game with each of its competitors (i.e., each potentially conflicting vehicle). If the competitor is CAV, the game will be a cooperative one. Otherwise, it is a non-cooperative game. In each game, each player may choose to be a follower or a leader. The corresponding expected acceleration and costs are calculated for each leader-follower combination, and the acceleration rates are computed by *Acceleration Control Module* to be introduced later. To dynamically adapt to the mixed traffic environment, a game is played in each time step, if a conflict exists. The game starts when the conflict emerges and ends until this conflict is solved.

e) Merging Sequence Determination Module: The merging sequence determination module is the last part of the game theory-based algorithm. The purpose of this module is responsible for coordinating the merging sequence dynamically by utilizing results from the game formation module. Each vehicle will obtain its role with respect to its competitor (i.e., leader or follower), as well as an optimal longitudinal acceleration that satisfies the safety constraints. If two competitors are both CAVs, they will share respective costs with each other and make a game-wise optimal decision together. Details of the game theory-based algorithm will be introduced in the next subsection.

f) Acceleration Control Module: This module is responsible for two main goals. The first one is to ensure ego vehicle can run at the desired longitudinal speed and track the lane. The second one is to perform the lane change maneuver safely, once the lane change condition is satisfied [132]. We formulate the vehicle longitudinal motion with a standard vehicle kinematic model as Equation 29, where the longitudinal acceleration is used as the control input.

$$\begin{bmatrix} s(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} \Delta t^2/2 \\ \Delta t \end{bmatrix} a(k) \tag{29}$$

where s, v and a represent the displacement, speed, and acceleration of the vehicle, respectively.

Once the ego vehicle confirms its target vehicle and the associated states, the consensus control algorithm [153] from our previous research is adopted to compute the acceleration. This allows the ego vehicle i to maintain a desired inter-vehicle gap and the same speed with its target vehicle j:

$$a_{ref}(k+1) = -\alpha_{ij}\beta_{ij} \cdot [(s_i(k) - s_j(k) + l_j + v_i(k) \cdot (t_{ij}^g(k) + \tau_{ij}(k))) + \gamma_i \cdot (v_i(k) - v_j(k)] \tag{30}$$

where α_{ij} denotes the value of adjacency matrix; β_{ij} and γ_i are control gains; $\tau_{ij}(t)$ denotes the time-varying communication delay between two vehicles; and $t_{ij}^g(t)$ is the time-varying desired time gap between two vehicles.

The string stability of this algorithm is well discussed in [153]. For a platoon of CAVs, this consensus-based control algorithm guarantees that the error signals are not amplified upstream along with the platoon, ensuring the string stability in a pure CAV environment.

4.3.2.3.2 Game theory-based merging sequence determination

During the merging process, complex conflict can be summarized with three types of scenarios, including the interactions between 1) two legacy vehicles, 2) two CAVs, and 3) a CAV and a legacy vehicle. This dissertation will only discuss the CAV(s) involved conflicts, since the conflicts between two legacy vehicles cannot be coordinated directly by CAVs. Hereafter, this dissertation will analyze the merging strategy from the perspective of the ego vehicle (CAV). Assumptions and specifications that are generally common in related literature are made as below:

- The proposed algorithm aims to control the longitudinal speed to provide a safe merging space, and low-level control of the steering angle is outside the scope of this dissertation.
- All CAVs that are involved in conflicts act cooperatively to achieve an optimal goal.
- The communication module and perception system of CAVs in the platform are assumed to be ideal. Therefore, no communication delay and packet loss are considered, and CAVs are capable of acquiring perfect information.
- The proposed algorithm guarantees the string stability within a pure CAV platoon instead of the whole mixed traffic flow. Since the string stability of mixing two car-following models is a complicated problem, it is still an open topic and out of our research interest.

4.3.2.3.2.1 Game formulation

When a potential conflict exists in the merging area, at least one of mainline vehicles and ramp vehicles needs to adjust its speed for a certain merging sequence. For the decision-making purpose, *Game Theory* is adopted for CAVs to evaluate their situation and then figure out the optimal merging strategy. A two-player non-zero-sum game is used in this dissertation to handle each conflict by providing a merging sequence for each player in the game.

In such a game, ego vehicle is named *Player 1* (hereafter "P1"), while its competitor is named *Player 2* (hereafter "P2"). Both P1 and P2 can choose either to be a leader or a follower, with the action set given as:

$A(P1) = \{1\text{: To be the leader}, 2\text{: To be the follower}\}$,

and $A(P2) = \{1\text{: To be the leader}, 2\text{: To be the follower}\}$.

The motivations of mainline vehicles and ramp vehicles may be different: mainline vehicles attempt to drive safely without compromising in travel speed, while ramp vehicles have to worry about the remaining distance to the end of merging area. As the remaining distance decreases, the merging intention of ramp vehicles may grow, and this anxiety can be expressed as the risk value in the cost function.

4.3.2.3.2.2 Cost function

Safety is always the first priority to be considered. For each action of ego vehicle in the game, a corresponding suggested acceleration $\hat{a}$ is calculated by the control algorithm. Therefore, we can predict the time-to-collision (TTC) in the next time step of each action.

From the perspective of ego vehicle, the predicted TTC for any pair of players can be formulated in Equation 31,

$$\hat{t}_{TTC} = \left[d_{gap} + \Delta\hat{d}_{gap}\right] / \left[v_f + \Delta v_f - \left(v_p - \Delta v_p\right)\right], \quad if\ v_f + \Delta\hat{v}_f > v_p - \Delta\hat{v}_p \quad (31)$$

where v_f, and $\Delta\hat{v}_f$ are the current speed and the predicted speed change of the following vehicle, respectively; v_p and $\Delta\hat{v}_p$ are the current speed and the predicted speed change of the preceding vehicle, respectively; d_{gap} is the current inter-vehicle gap; $\Delta\hat{d}_{gap}$ is the predicted gap change.

In the "two CAVs" scenario, these predicted values in Equation 31 are shared with each player because of the vehicle communication. For the "a CAV and a legacy vehicle" scenario, the predicted values of legacy vehicles can be assumed to be unchanged during a small time interval Δt (0.02s in the simulation), allowing CAV to estimate legacy vehicles' action. For example, in Equation 31, if the following vehicle is a legacy vehicle, the predicted speed change $\Delta\hat{v}_f = 0$.

However, using only TTC is not enough to quantify the safety risk [154]. For example, if the preceding vehicle is faster than the following vehicle, TTC will be negative. Moreover, if the difference between the following vehicle and the preceding vehicle is small, it will generate a huge TTC indicating safety even though the inter-vehicle gap is small. By combining predicted TTC ($\hat{t}_{TTC}$) and predicted time headway of ego vehicle ($\hat{h}_{ev}$). The cost of rear-end collision risk (J_{risk}^{c}) for each action can be evaluated below as in Equation 32:

$$J_{risk}^{c} = \begin{cases} \left(\begin{matrix} [1 - tanh(\hat{t}_{TTC}/H_{min})] + \\ [1 - tanh(\hat{h}_{ev}/H_{min})] \end{matrix} \right)/2, & \hat{t}_{TTC} \geq 0 \\ [1 - tanh(\hat{h}_{ev}/H_{min})]/2, & \hat{t}_{TTC} < 0 \end{cases} \quad (32)$$

$$\hat{h}_{ev} = [d_{gap} + \Delta\hat{d}_{gap}]/(v_f + \Delta\hat{v}_f) \quad (33)$$

where H_{min} is the minimum safe time headway based on the 3-second rule [129], $\hat{h}_{ev}$ is the predicted time headway of ego vehicle.

To consider the merging urgency of a ramp vehicle, the distance to the end of merging area should be added to the risk value of ramp vehicle, as shown in Equation 32. The closer to the end of the merging area, the higher cost vehicles should pay. The risk of merging (J_{risk}^{m}) can be formulated as follows:

$$J_{risk}^{m} = [1 - tanh(\hat{h}_{rv}/H_{min})]/2 \quad (34)$$

$$\hat{h}_{rv} = [d_r + \Delta\hat{d}_r]/(v_e + \Delta\hat{v}_e) \quad (35)$$

where $\hat{h}_{rv}$ is the predicted remaining time headway to the end of merging area for ramp vehicle; d_r is the remaining distance of merging area; $\Delta\hat{d}_r$ is the predicted remaining distance; v_e and $\Delta\hat{v}_e$ are the current speed and predicted speed change of ego vehicle, respectively.

To summarize, the risk for mainline vehicles and ramp vehicles used in this study can be expressed in Equation 36:

$$J_{risk} = \begin{cases} J_{risk}^{c}, & Mainline\ vehicles \\ (J_{risk}^{c} + J_{risk}^{m})/2, & Ramp\ vehicles \end{cases} \quad (36)$$

As aforementioned, saving travel time may be another target for players in the game. In addition, if we only consider safety in the cost function, players in the game will incline to more conservative behaviors (e.g., encouraged to be followers or to decelerate), resulting

in unnecessary congestion along with the upstream. Therefore, adding a mobility term would help CAVs find the balance between safety and speed, and improve the traffic efficiency at the same time. Both mainline and ramp CAVs are encouraged to take actions with minimum speed drop, if the safety performance is not compromised. As shown in Equation 37, the mobility cost function puts more penalties on deceleration maneuvers. The term $tanh(\Delta\hat{v}_e\ /v_e)$ is more sensitive when the speed of ego vehicle is slow, as this algorithm cares more about mobility for low-speed driving, but safety for high-speed driving.

$$J_{mobility} = 1 - tanh(\Delta\hat{v}_e\ /v_e) \tag{37}$$

where $\Delta\hat{v}_e$ is the ego vehicle speed difference of either being a follower or a leader in the game, compared to the current speed.

To improve the driving comfort, hard braking and drastic acceleration are penalized with a cost as shown in Equation 38.

$$J_{comfort} = \begin{cases} \hat{a}/acc_{lim}\ , & \hat{a} \geq 0 \\ \hat{a}/dec_{lim}\ , & \hat{a} < 0 \end{cases} \tag{38}$$

where $\hat{a}$ is the acceleration of ego vehicle in next time step, $acc_{lim} > 0$ is the acceleration limit, and $dec_{lim} < 0$ is the deceleration limit.

To summarize, the overall cost ($\bar{J}$) is:

$$\bar{J} = \alpha_1 J_{risk} + \alpha_2 J_{mobility} + \alpha_3 J_{comfort} \tag{39}$$

where $\alpha_i \geq 0, i = 1,2,3$, is the weight for each term in the cost function, and $\sum_i \alpha_i = 1$. In this study, we choose $\alpha_1 = 0.4$, $\alpha_2 = 0.4$ and $\alpha_3 = 0.2$. A more detailed sensitivity analysis is presented in the Section 4.3.2.4.4.

4.3.2.3.2.3 Non-cooperative game and cooperative game

After estimating the cost of each player's action, the optimal result can be obtained from a decision table, which depends on the game type, either non-cooperative or cooperative game.

In this study, a game can be only initiated by an equipped CAV. Once the CAV recognizes a potential conflict, it sends out a cooperation invitation and keeps waiting for a reply. If the CAV receives no response from the other party, a non-cooperative two-player game will be formed. In this type of game, the CAV will adopt a selfish strategy, since it can only rely on the information from the radar system and optimize its own cost. The decision table of the non-cooperative game is shown in Table IX.

TABLE IX DECISION TABLE FOR NON-COOPERATIVE TWO-PERSON GAME

	Competitor		
Ego vehicle	Role	Leader	Follower
	Leader	∞	$\bar{J}_{lead}$
	Follower	$\bar{J}_{follow}$	∞

To avoid collision, ego vehicle will not choose to play the same role with its competitor at the same time. Therefore, the costs for both players being the leaders or followers simultaneously are set to be infinite (or very large values). At each time step, ego vehicle will choose the option with the minimum expected cost, as described in Equation 40.

$$Action = \min_{actions} \{\bar{J}_{lead}, \bar{J}_{follow}\} \quad (40)$$

The game between two CAVs would be a cooperative one, where players can make decisions together. The decision table of the cooperative game between two CAVs is shown in X. Unlike a non-cooperative algorithm which can provide the optimal solution only for ego vehicle regardless of system conditions, a cooperative game can optimize the total cost (based on the information shared via vehicle-to-vehicle communication) for both CAVs.

TABLE X DECISION TABLE FOR COOPERATIVE TWO-PERSON GAME

	Partner		
Ego vehicle	Role	Leader	Follower
	Leader	∞	$\bar{J}_{lead}^{ego} + \bar{J}_{follow}^{p}$
	Follower	$\bar{J}_{follow}^{ego} + \bar{J}_{lead}^{p}$	∞

As described in Equation 41, both CAVs will take the action to achieve the system optimum.

$$Action = \min_{actions} \{\bar{J}_{follow}^{ego} + \bar{J}_{lead}^{p}, \ \bar{J}_{lead}^{ego} + \bar{J}_{follow}^{p}\} \quad (41)$$

where $\bar{J}_{follow}^{ego}$ and $\bar{J}_{lead}^{ego}$ are the costs of being a follower or a leader for ego vehicle, respectively; $\bar{J}_{follow}^{p}$ and $\bar{J}_{lead}^{p}$ are the costs of being a follower or a leader for its partner, respectively.

4.3.2.3.2.4 Disturbance filtering

Although our system is designed to adapt to any environment dynamically, the game result should not be too sensitive to the small environmental disturbance. Hence, the hysteresis controller [152] is adopted to filter the game results and prevent unwanted rapid

state switching as aforementioned. By taking recent system history into account, hysteresis can filter signals so that the output reacts less rapidly.

As shown in Fig. 4-37, the system history can be logged by a counter, ranging from -*N* to *N*, which will increase when the cost of being the leader is greater than being the follower, which will increase/decrease when the cost of being the leader is greater/smaller than being the follower. Specifically, when the counter is greater than *n*, the output action is "*being the follower*". It will not switch to being the leader immediately when the counter drops below *n*. Only when the counter decreases below -*n*, the output action will switch to being the leader.

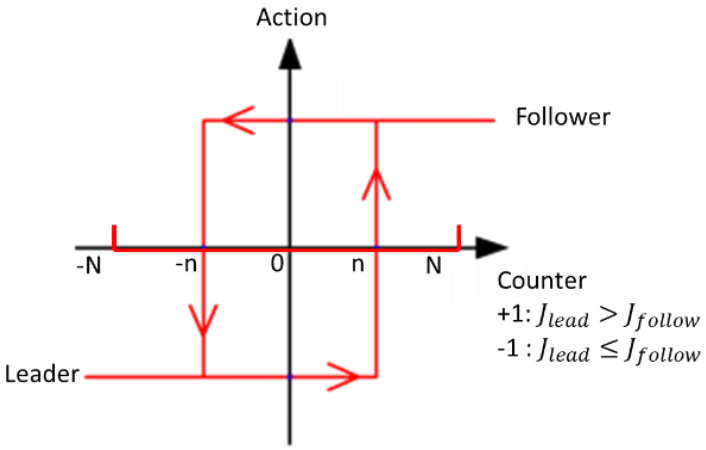


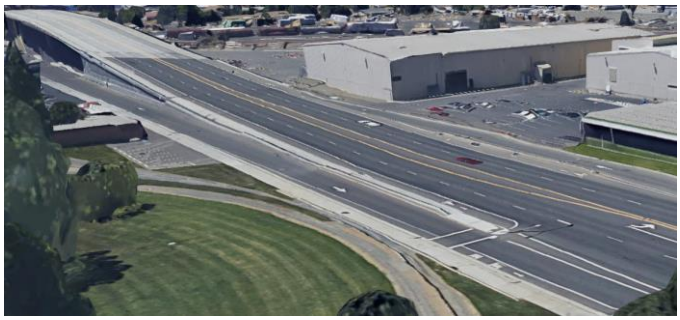
Fig. 4-37 Hysteresis controller for disturbance filtering.

4.3.2.4 Case Study and Results Evaluation

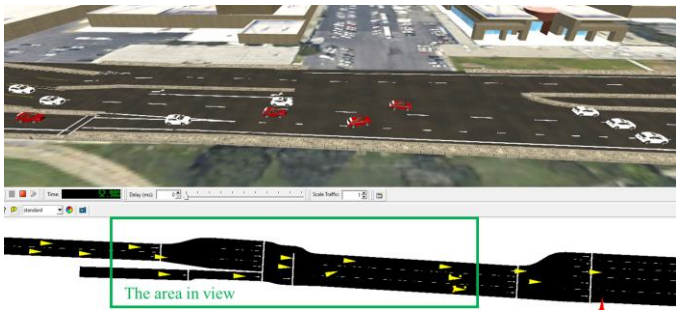
On the customized Human-in-the-Loop Co-simulation Platform in Section 4.1.2, a traffic flow level simulation is carried out under different CAV penetration rates and congestion levels. The simulation results are analyzed in terms of safety, mobility and fuel efficiency.

4.3.2.4.1 Unity-SUMO co-simulation

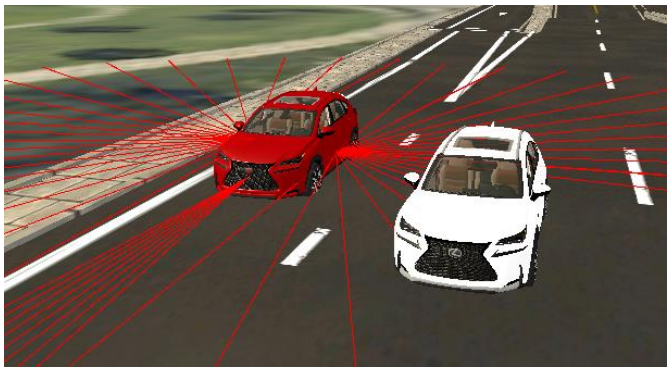
A real-world traffic network is coded in the simulation, spanning from the intersection of Chicago Avenue to the intersection of Iowa Avenue along Columbia Avenue in Riverside, California. It consists of a single-lane on-ramp and a segment of multi-lane mainline (Google Maps view is shown in Fig. 4-38(a). The integrated simulation environment is shown in Fig. 4-38(b), where the upper part with terrain details is the Unity environment, and the lower part is the corresponding SUMO network. SUMO controls legacy vehicles while Unity controls CAVs with the proposed algorithm. CAVs are color-coded in red, while legacy vehicles are in white. As shown in Fig. 4-38(c), the red rays spread from the CAV indicate the detection range of its onboard radar system.



(a)



(b)



(c)

Fig. 4-38. Unity-SUMO integrated simulation based on a real-world ramp merging area in Riverside, CA: (a) View from Google Maps at the real-world ramp; (b) User interface of the Unity-SUMO co-simulation platform; (c) A CAV (in red) with radar system and a legacy vehicle (in white).

4.3.2.4.2 Simulation design

Default car-following and lane-changing models of SUMO are used to generate the legacy traffic flow as the baseline. Then, the proposed algorithm is evaluated in a mixed traffic simulation under different penetration rates.

To generate a more realistic mixed traffic environment and carry out a fair evaluation, the parameters are carefully selected as shown in TABLE III. In addition to the car-following and lane change model discussed in the platform introduction Section 4.1.2, more details are elaborated below.

To be specific, both the desired time headway and reaction time is set to be 1 second, based on the Krauss parameters calibration [155] and real-world data fitting [156]. To generate the traffic flow as close to the real-world situations as possible in SUMO, Bjärkvik et al. [157] well-tuned the parameters of the Krauss car-following model, such as the minimum desired time headway (*tau*), acceleration, deceleration, and the result was verified with collected traffic data in the real-world. In this study, the speed limit of 20m/s is chosen based on the speed limit of the specific road segment of the testbed. As a result, parameter distributions of the car-following model in our simulation are well aligned with the suggested values in Bjärkvik et al.'s work. Moreover, the car-following model of CAV comes from the one in our previous study, as shown in Equation 30.

As shown in Fig. 4-38(b), vehicles are running on a two-lane mainline segment and a ramp, so mainline vehicles on the right lane can avoid the conflict with ramp vehicles by changing to the left lane. As Table XI shows, the proposed algorithm is evaluated in three congested levels, including light traffic with the volume-to-capacity (V/C) ratio of 0.35, moderate traffic with the ratio of 0.6, and congested traffic with the ratio of 0.85, where traffic demands are 1400, 2400 and 3400 vehicles per hour, respectively. The ratio of ramp demand to mainline demand is 1:2. In addition, to assess the system performance of the proposed algorithm in various mixed traffic scenarios, four levels of CAV penetration rate

(i.e., 0%, 30%, 70%, and 100%) are evaluated in the simulation. Moreover, to statistically analyze the simulation results, 3 random seeds are selected for each of 12 scenarios (i.e., 3 congestion levels and 4 penetration rates). Therefore, a total of 36 simulation runs are carried out in this study. Based on these preset traffic demands, a traffic stream generator schedules the itinerary for each vehicle, which is modeled as a Poisson process. Therefore, vehicles' departure times follow an exponential distribution. The simulation only terminates when the last vehicle leaves the network to guarantee fair comparison across different strategies.

TABLE XI SIMULATION SETUP PARAMETERS

Vehicular Parameters		
Vehicle type	CAVs	Legacy vehicles
Control platform	Unity	SUMO
Initial speed (adaptive to traffic)	ramp: 15 m/s; mainline: 20 m/s	
Minimum inter-vehicle gap	5 m	
Acceleration range	-5 ~ 3 m/s^2	
Desired speed (speed limit)	20 m/s	
Desired minimum time headway	1 s	
Initial distance to merging point	ramp: 250 m; mainline: 280 m	
Emergency braking	-9 m/s^2	
Simulation Environment Parameters		
Penetration rate of CAVs	0%, 30%,70%,100%	
Congestion level (v/c ratio)	0.35, 0.60, 0.85	
Traffic demand (veh/hr)	1400, 2400, 3400	
Merging zone length	89 m	
Speed limit	20 m/s	
Simulation time step	0.02 s	

4.3.2.4.3 Simulation result analysis

To evaluate the performance of the proposed algorithm, we analyze the processes for two types of games, and their influence on the traffic flow. For the traffic flow level evaluation,

mobility, energy efficiency and driving volatility are analyzed in each traffic scenario for mainline and ramp vehicles, respectively.

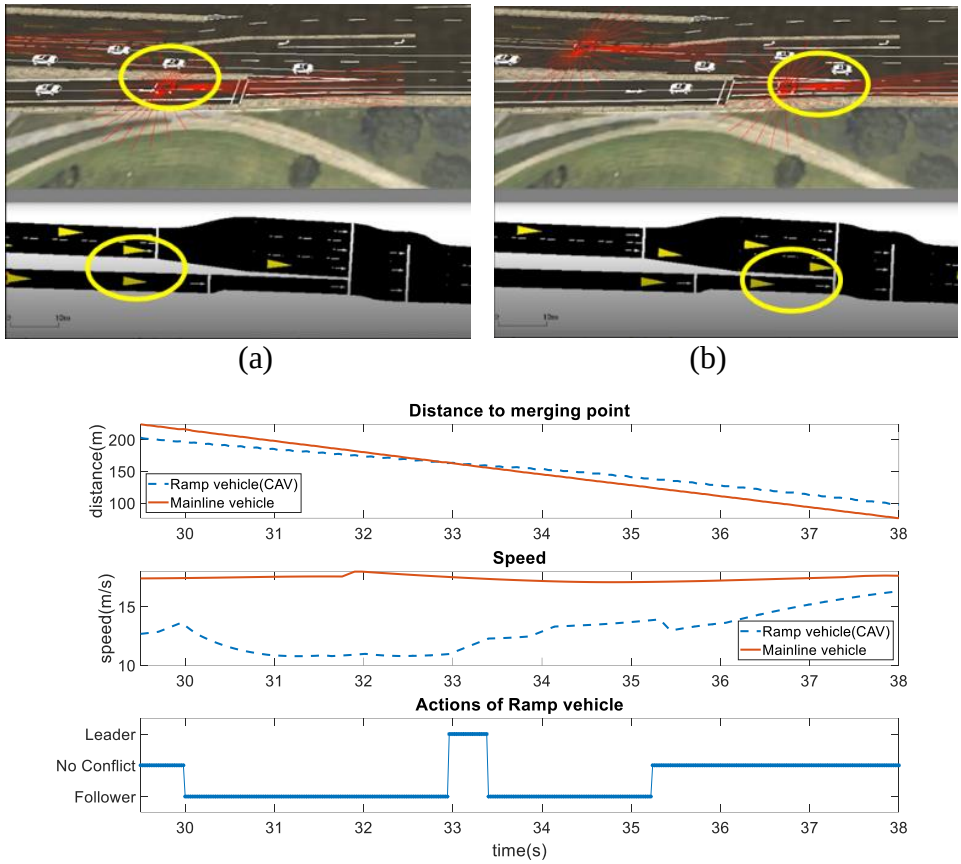


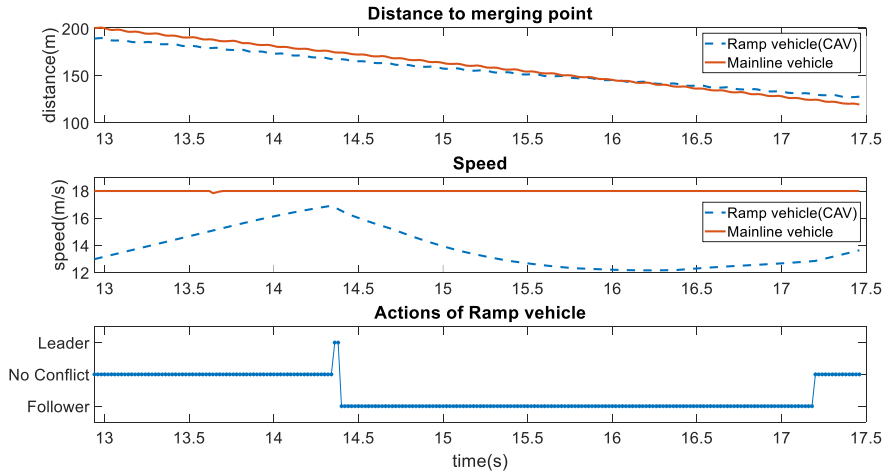
Fig. 4-39 Game process of a non-cooperative game: (a) An on-ramp CAV and a mainline legacy vehicle compete for merging; (b) Merging order is determined; (c) the whole process of the game.

4.3.2.4.3.1 Game process

Two merging conflicts are selected from the simulation for illustrating how the game evolves. Fig. 4-39 shows an example of a non-cooperative game between a CAV and a legacy vehicle. In Fig. 4-39 (a), as circled in yellow, an on-ramp CAV (in red) encounters a mainline legacy vehicle (in white), and a non-cooperative game is formed. In Fig. 4-39 (b), the game is settled with the merging order determined, and the CAV finds a slot with

ensured safe longitudinal merging gap. Fig. 4-39 (c) presents the whole process of the game, starting from 29.98s and ending at 35.24s. The CAV first decides to be the follower when the conflict emerges. At 32.96s, two vehicles drive in parallel, and when the legacy vehicle slows down, CAV decides to be the leader. However, the legacy vehicle does not mean to yield and keeps running at a high speed. Therefore, CAV's decision flips back to being a follower.

(a) (b)



(c)

Fig. 4-40 Game process of a cooperative game: (a) Two CAVs compete for merging; (b) Merging order is determined; (c) the whole process of the game.

Fig. 4-40 shows how two CAVs solve a similar conflict with a cooperative game. The actions of two CAVs are exclusive, with one being the follower and the other one being

the leader. In Fig. 4-40 (c), before the conflict starts, the on-ramp CAV accelerates to reach the mainline speed. At the instant of two CAVs encountering each other, the merging sequence is decided. In case of any unexpected emergency braking of one CAV due to preceding legacy vehicles, however, the game still exists until the CAV solves its conflicts or leaves the merging area. Compared with the non-cooperative game, the decision of two CAVs is stable, and the mainline vehicle does not need to change its speed. Moreover, the cooperative game takes only 2.86 s to solve the conflict, which is much faster than 5.26s in non-cooperative game.

4.3.2.4.3.2 Mobility

Fig. 4-41 (a) and Fig. 4-41(b) present the average speeds (i.e., the ratio between vehicle-meter-traveled and vehicle-second-traveled) of mainline vehicles and ramp vehicles, respectively. The error bars (one standard deviation) indicate the result variability. As shown in the figures, the variance is reduced when the penetration rate grows, or the congestion level decreases. It can be observed that the proposed algorithm improves not only the average speed of mainline vehicles, but also the one of ramp vehicles.

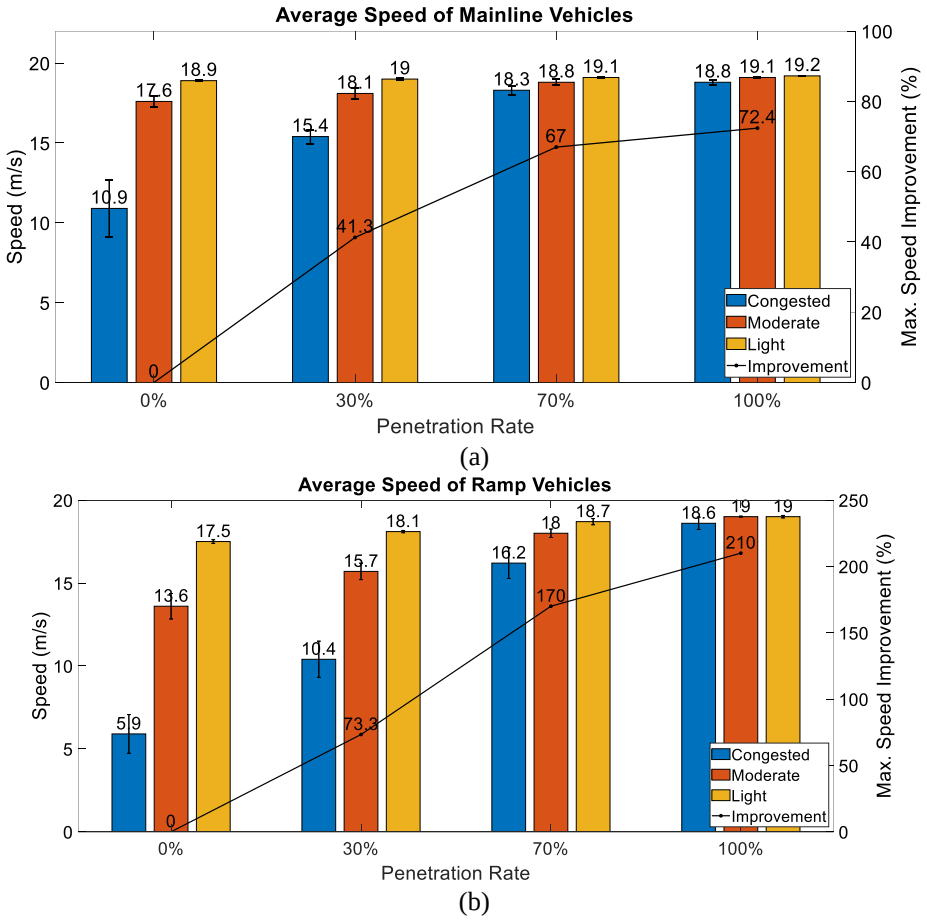


Fig. 4-41 Speed of traffic flows: (a) Average speed of mainline vehicles; (b) Average speed of on-ramp vehicles.

In Fig. 4-41, the solid line indicates the most significant improvement made by the proposed algorithm, compared with the baseline (i.e., 0% penetration rate). The largest improvement of average speed is gained in congested traffic, i.e., 210% for ramp vehicles and 72.4% for mainline vehicles, compared with the baseline. When the penetration rate grows, the average speed increases significantly because the proposed algorithm for each CAV can coordinate the merging maneuvers (including the sequence) implicitly, which helps mitigate the congestion. It is noted that in scenarios with 100% penetration rate of

CAVs, average speeds for both mainline vehicles and ramp vehicles are close to the free flow speed (20 m/s), regardless of congestion levels. This means that the proposed algorithm can effectively regulate the traffic under different traffic demands when the penetration rate of CAVs is high.

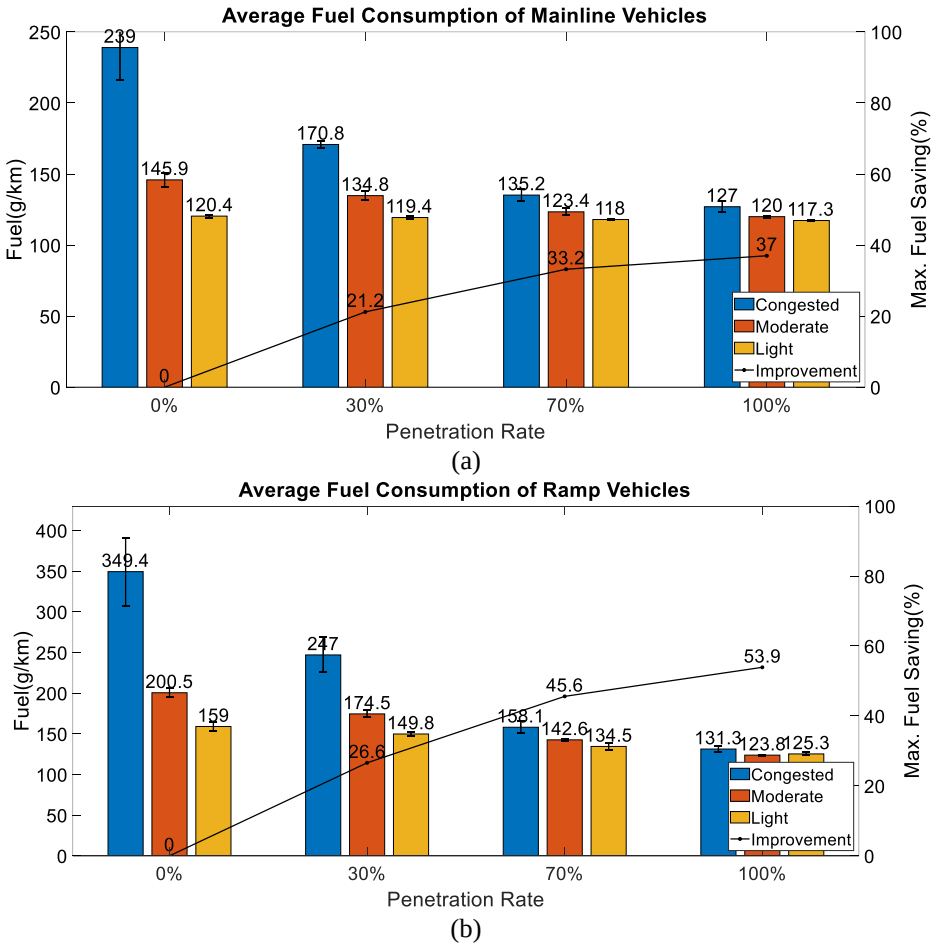


Fig. 4-42 Average fuel consumption: (a) Fuel consumption of mainline vehicles; (b) Fuel consumption of on-ramp vehicles.

4.3.2.4.3.3 Energy efficiency

We also analyze the energy efficiency of the proposed strategy with the open-source MOVESTAR fuel and emission model [158]. Fig. 4-42(a) and Fig. 4-42(b) present the fuel

consumption of mainline vehicles and ramp vehicles, respectively, under the assumption that all vehicles in the simulation are passenger cars powered by gasoline. The variance of energy results shows the same pattern as the speed results. As expected, the results show that the proposed algorithm can considerably reduce the fuel consumption for both mainline and ramp vehicles, where the most significant reduction of fuel consumption happens in the congested traffic scenario for ramp vehicles. As the improvement in Fig. 4-42 shows, up to 53.9% of fuel consumption can be saved since vehicles can merge in a coordinated manner and minimize speed changes that might generate shock waves.

4.3.2.4.3.4 Driving volatility

Driving volatility is defined as the deviation from the norm, reflecting the stability of the vehicle's movement. As the higher driving volatility is associated with higher risk [159], it can also be used as one of the surrogate measures for safety. In this study, we adopt the "*percent of extreme values*" method to evaluate the speed volatility which can quantify the driving risk and comfort. This performance index captures driving volatility by counting the number of observations beyond a defined threshold-band, where any hard brake or drastic acceleration will increase the volatility.

The speed volatility ($V\%$) can be defined in Equation 42:

$$V\% = \frac{1}{t}\sum_{i=0}^{t-1} I\{x(i) > \mathrm{T}_{upper}, x(i) < \mathrm{T}_{lower}\} \quad (42)$$

where $I\{\cdot\}$ denotes the indicator function, which equals 1 if the mathematical expression "statement" is true, and otherwise, it equals 0; $x(i)$ is the observed value at time step i; and

t is the total time steps of the observations. The upper and lower thresholds, T_{upper} and T_{lower}, can be defined in Equation 43:

$$\begin{cases} T_{upper} = \overline{x} + 2 \times S_{dev} \\ T_{lower} = \overline{x} - 2 \times S_{dev} \end{cases} \tag{43}$$

where $\overline{x}$ is the mean of observations, and S_{dev} is the standard deviation.

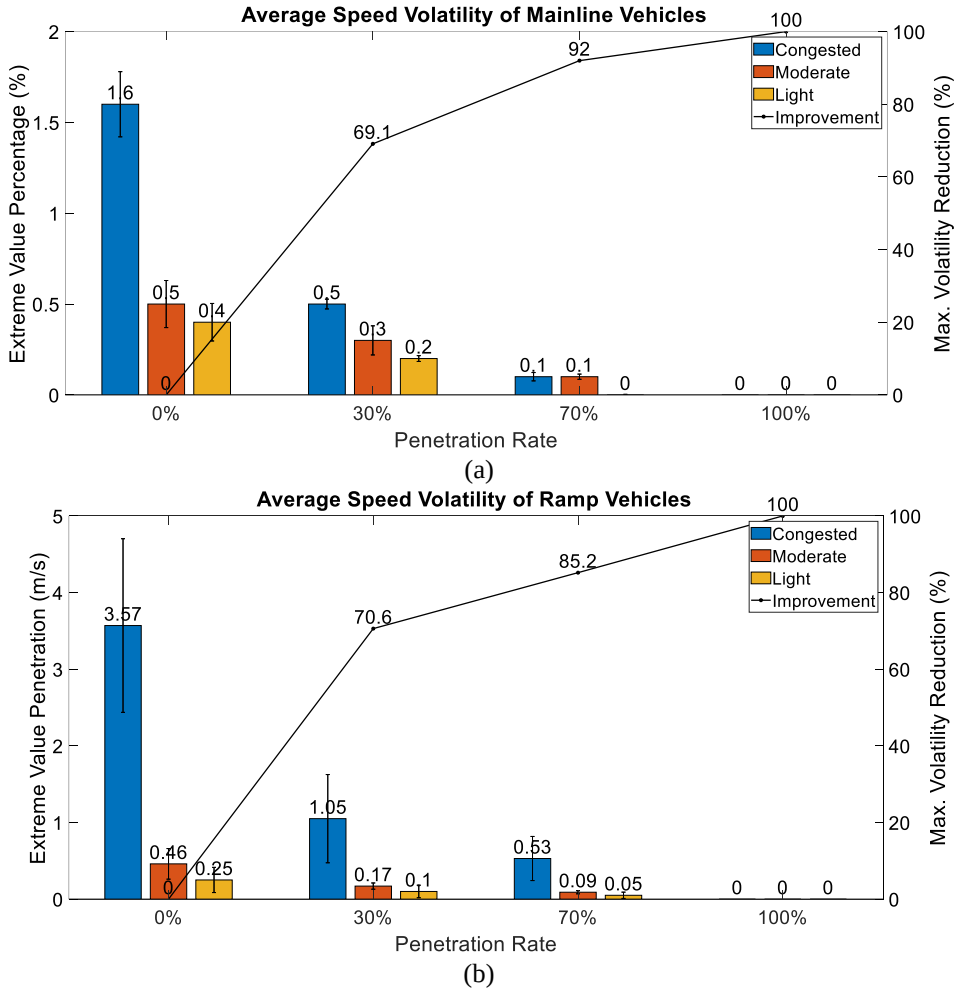


Fig. 4-43. Average driving volatility: (a) Speed volatility of mainline vehicles; (b) Speed volatility of on-ramp vehicles.

Fig. 4-43(a) and Fig. 4-43(b) show the speed volatility of mainline and ramp vehicles, respectively. As the CAV penetration rate increases, the speed volatility decreases

significantly. The most remarkable improvement is obtained in 100% CAV penetration rate scenarios, where the outliers or extreme values (due to hard brake or drastic acceleration) are completely eliminated.

4.3.2.4.4 Sensitivity analysis on weights

In cost functions Equation 39, the weights α_1, α_2 and α_3 represent how much the algorithm would value safety, mobility, and driving comfort, respectively. Tuning these weights can generate different driving behaviors for CAVs, thus affecting the traffic flow in different ways. To better understand the impacts of these weights, we carry out simulations for 10 different combinations (see Fig. 4-44), and analyze the results of mobility, energy efficiency, and driving volatility, respectively. Because $\sum_i \alpha_i = 1$ and $\alpha_i \geq 0$, we set $\alpha_3 = 1 - \alpha_1 - \alpha_2$, and select α_1 and α_2 in the domain of $\alpha_1 + \alpha_2 - 1 \leq 0$. As shown in Fig. 4-44, 10 red dots representing 10 different weight combinations are evenly distributed within the domain.

TABLE XII SENSITIVITY ANALYSIS FOR CAV PERFORMANCE MEASUREMENT

α_1	α_2	α_3	Avg. Fuel (g/km)	Avg. Speed (m/s)	Speed Volatility (%)	Acc. Volatility (%)
0.125	0.125	0.75	130.22	18.71	0	4.5
0.375	0.125	0.5	**124.31**	18.69	0.02	**4.44**
0.625	0.125	0.25	127.37	18.67	0.02	4.84
0.875	0.125	0	127.03	18.69	0.07	5.06
0.125	0.375	0.5	129.84	18.59	0.06	**4.46**
0.375	0.375	0.25	**125.18**	**18.91**	0.03	**4.49**
0.625	0.375	0	126.98	18.76	0	4.84
0.125	0.625	0.25	126.05	**18.83**	0	4.55
0.375	0.625	0	**125.12**	**18.83**	0.03	4.71
0.125	0.875	0	125.34	**18.88**	0.05	4.85

We analyze the direct influence of weight tuning only for CAVs rather than the whole traffic flow, because the weights determine the behavior of the CAV, and the influence of the weight on legacy vehicle is indirect. Simulation results of sensitivity analysis are shown in Table XII and Fig. 4-45. The fuel consumption varies from 125.83 to 131.64 g/mi. The top 3 lowest values are located in area 2, 6 and 9 (Fig. 4-44), with the weight combinations of (0.375, 0.125, 0.5), (0.375, 0.375, 0.25) and (0.375, 0.625, 0), respectively. According to the results in Fig. 4-45(a), the average fuel consumption shows a decreasing trend as α_1 and α_2 grow, and at $\alpha_1 = 0.375$, it reaches a local minimum.

As Fig. 4-45(b) shows, the average speed increases as α_2 increases since α_2 is the weight for mobility. The speed volatility ranges only from 0 to 0.07%, which is trivial. Thus, the acceleration volatility is further considered. As Fig. 4-45(c) shows, in the area where both α_1 and α_2 are small, the value of acceleration volatility is small, since the cost function emphasizes more on the comfort term. As $\alpha_3 = 1 - \alpha_1 - \alpha_2$ decreases, the acceleration volatility increases.

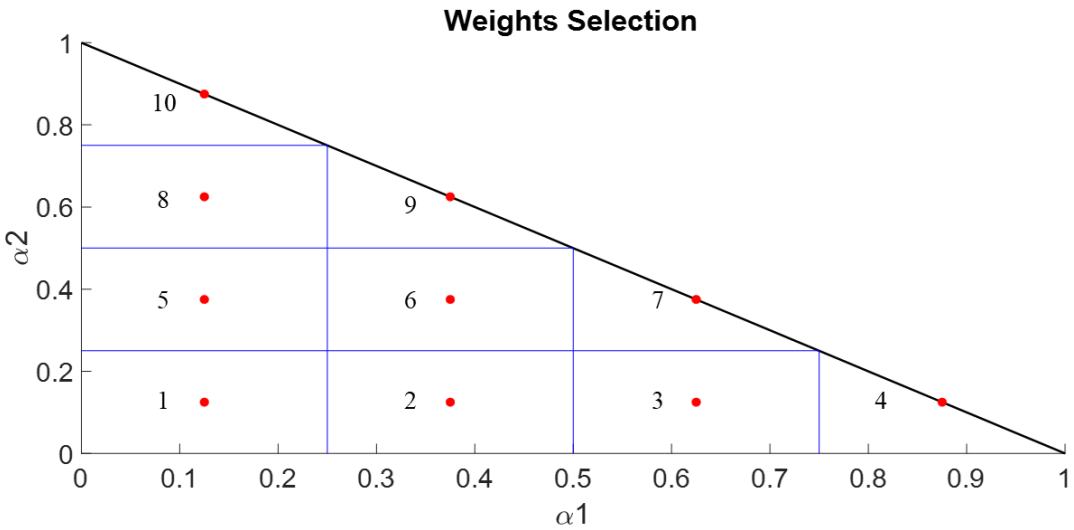
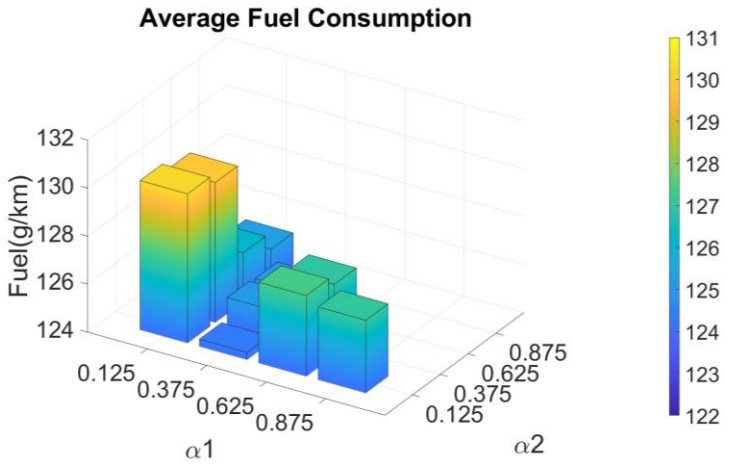
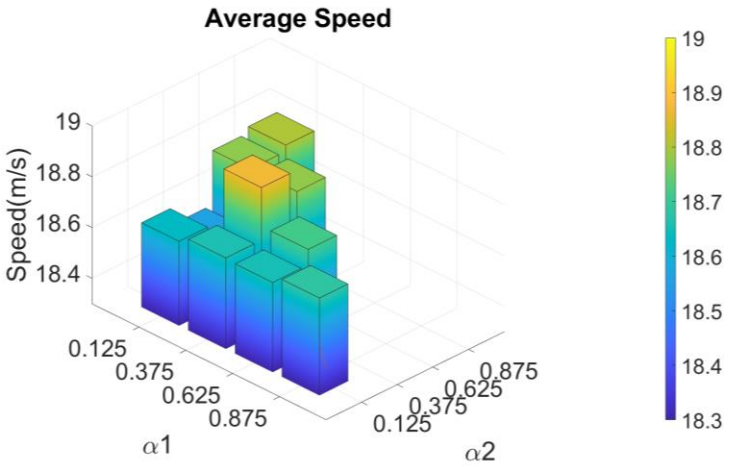


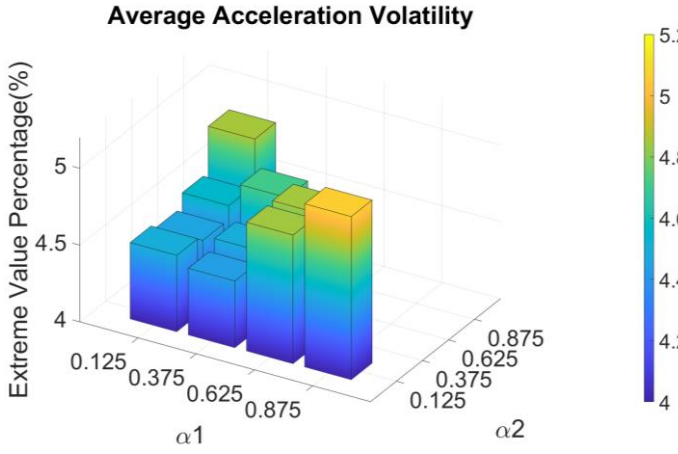
Fig. 4-44 Weight selection for sensitivity analysis



(a)



(b)



(c)

Fig. 4-45 Sensitivity analysis for CAV performance measurement: (a) Average fuel consumption, (b) average speed, and (c) average acceleration volatility.

4.3.2.5 Conclusions and Future Work

In this study, a game theory-based ramp merging strategy has been proposed for CAVs in the mixed traffic environment. The system has been developed, implemented, and evaluated in a customized co-simulation platform, which fuses a game engine-based simulator (Unity) and a microscopic traffic simulator (SUMO). Compared with the baseline merging algorithm of SUMO, the proposed algorithm can significantly improve system mobility (by up to 210%) and reduce fuel consumption (by up to 53.9%) under different traffic demands and CAV penetration rates.

As one of the few ramp merging algorithms developed for mixed traffic, some challenges need to be addressed along its future development pathway: 1) Human-machine interaction in the game theory-based algorithm needs further investigation, since the modeling of human behaviors is still challenging; 2) With the development of Unity-SUMO co-simulation platform, human-in-the-loop (HuiL) simulation tests can be carried out to obtain valuable data for more in-depth human behavior research; and 3) In addition to safety, mobility and driving comfort, environment-related factors can be considered in the design of game functions to build an eco-friendly ramp merging system.

4.3.3 Driving Behavior Adaptive Advanced Driving Assistance System

4.3.3.1 Introduction and Background

4.3.3.1.1 Motivation

In the rapidly evolving world of transportation, the presence of intelligent vehicles (IV) on the road is increasing. However, human-driven vehicles still hold a significant market share. To build a harmony traffic system, empowering human-driven vehicles with

advanced technologies has become a significant goal. By extending the algorithms of intelligent vehicles, we can empower human drivers to excel like never before. As the previous chapter discussed, these advanced algorithms analyze data, model driving behavior, adapt to situations, and enhance decision-making. They enable drivers to navigate confidently, learn from each experience, and contribute to a more efficient and collective transportation system. Let us embrace this opportunity to unlock the true potential of human-driven vehicles and shape a brighter, empowered future on the road.

As an enabler of intelligent vehicles, advanced driver-assistance systems (ADAS) aim to support drivers by either providing warnings or advisory information to reduce risk exposure, or to relieve drivers' burden by automating some of the driving tasks[160]. Many studies with respect to ADAS have been conducted over the past decades, such as pedestrian detection [161], vehicle overtaking [162], forward collision avoidance, and adaptive cruise control [163].

Advisory speed assistance (ASA) is a typical example of ADAS, which usually creates recommended speed limits to bound the vehicle speed on highways, or generates suggested speed profiles for the human driver to track on signalized arterials [164], [165]. The advisory information is delivered to the driver via a driver-vehicle interface (DVI), allowing him/her to control the longitudinal speed profile to gain safety, mobility, and/or environment benefits.

However, such ASA systems inevitably introduce human tracking errors, since it is impossible for a driver to track the suggested speed perfectly. It was concluded by one of our previous studies that, the human tracking errors may contribute to as high as 12%

degradation in system performance of the target ASA [166]. Therefore, how to model an individual driver's driving behavior and how to compensate for them in a customized way for advisory information delivery are critical to the design of an ASA system.

4.3.3.1.2 Summary and contribution

In summary, this section presents an innovative approach to extending the algorithms of intelligent vehicles to human-driven vehicles through the development of an Advisory Speed Assistance (ASA) system. The ASA system utilizes the game theory-based ramp merging coordination algorithm to coordinate traffic near on-ramp merging areas. Instead of relying fully automation, the ASA system provides speed guidance to human drivers through a head-up display (HUD) for human driven vehicles. To enhance the effectiveness of the ASA system, a learning-based approach is proposed to build a driver behavior model, predict human tracking errors, and compensate the speed suggestion. The ASA system is evaluated using the HuiL (human-in-the-loop) co-simulation platform, allowing volunteer drivers to participate in simulations and provide feedback on HUD design. Furthermore, the ASA system is implemented and tested in real-world scenarios using the vehicle-edge-cloud test bed. The results of this research contribute to the advancement of intelligent transportation systems and the integration of human drivers with intelligent algorithms.

Compared to many previous studies on driver behavior modeling, the major contributions of this study are listed below:

- Personalized advisory: By adopting the learning-based approach with a nonlinear autoregressive (NAR) neural network, the proposed system can classify different drivers into certain types and provide personalized advisory information. To this end, the

accuracy and effectiveness of the proposed ASA system can be greatly improved, compared to existing systems that only provide general advisory information to all drivers.

- Integrated vehicle system: Instead of only focusing on predicting driver behavior, we designed an integrated vehicle system which includes a motion planner, controller, a learning model, and user interfaces (both HUD and HMI). This integrated vehicle system is also presented by two proofs of concept.
- Multi-platform online validation: Different from some existing work that validate their driver behavior models using synthetical data with numerical simulation, we design an ASA system with HUD in the game engine and invite various volunteer drivers to conduct the human-in-the-loop simulation on the driving simulator platform, so the improvement of the proposed driver behavior model can be observed in real time. Furthermore, we validate the effectiveness of our driver behavior model using a real passenger vehicle. This study allows various drivers to participate and test the proposed model on two different platforms, so the results will be more convincing than any previously proposed studies which simply conduct computer simulation.
- Wide applicability: Although only a specific cooperative merging scenario is considered in this study, the proposed learning-based driver behavior model can be applied to various traffic scenarios, as long as the advisory speed is helpful for human drivers.

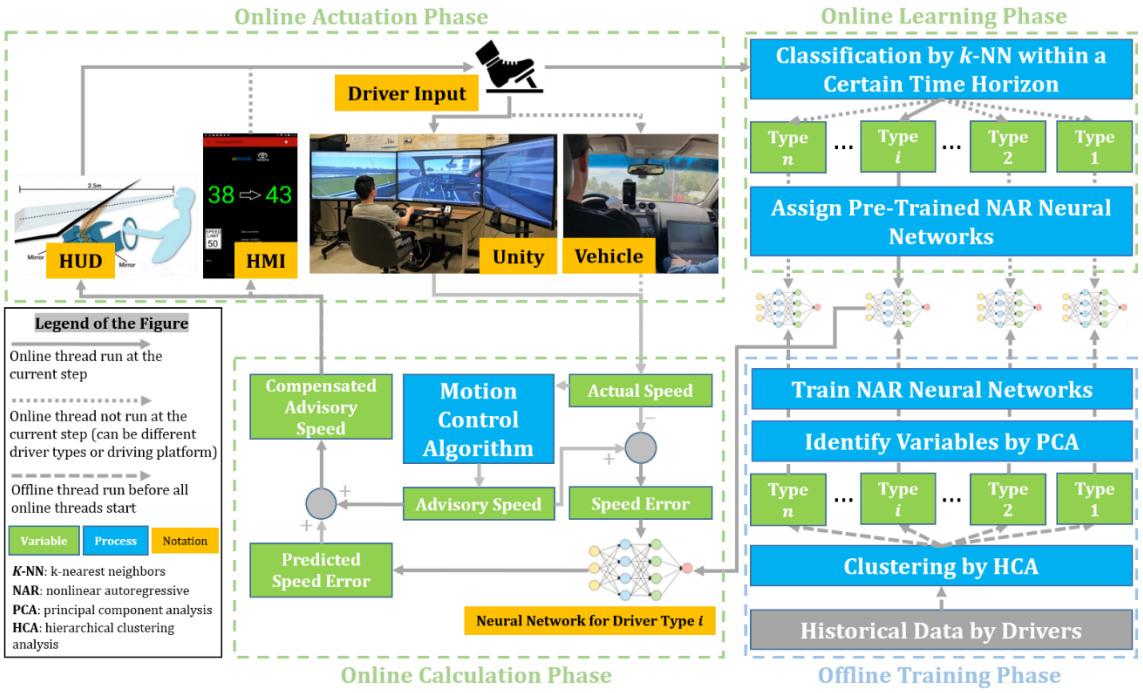


Fig. 4-46 System architecture of the proposed learning-based driver behavior modeling system

4.3.3.2 Methodology

4.3.3.2.1 Overview

The general architecture of the driving behavior ASA is shown in Fig. 4-46. The whole system can be broken down into four different phases, including the online actuation phase, the online calculation phase, the online learning phase, and the offline training phase. In the offline training phase, by learning the behavior of different drivers based on historical data, we cluster drivers into N types and train N neural networks. In the online learning phase, a driver will be classified into one of the preset types and assigned to the associated neural network according to his/her driving style within a certain time horizon. In the closed loop, once the neural network takes the speed tracking errors as inputs, it predicts the driving error in the next time step. Then the advisory speed calculated by the merging

algorithm can be compensated by the error prediction module, thus shown on HUD, as presented in Fig. 4-47.

(a) Suggested speed on head up display (HUD) in simulation

(b) A driver is tracking suggested speed in a real passenger vehicle

Fig. 4-47 Human machine interface of speed suggestion in simulation and real-world vehicle

For use case, the Advisory Speed Assistance (ASA) system proves invaluable in improving driver behavior during the chaotic process of ramp merging. By providing speed guidance and coordinating drivers' actions, ASA ensures smoother and safer merging maneuvers, reducing the likelihood of accidents and traffic congestion. During a ramp merging, once the merging sequence is determined by Equation 40 or 41, the reference

acceleration a_{ref} for each driver is calculated by Equation 30. The advisory speed displayed on ASA system can be computed as:

$$v_i(t + \delta t) = v_i(t) + a_{ref}(t + \delta t) \cdot \delta t \tag{44}$$

where $v_i(t + \delta t)$ is the advisory speed shown to the driver, and $v_i(t)$ is the current speed of the ego vehicle.

While the driver cannot follow the exact speed suggestion, we need to model and predict the driver's tracking error for improving the speed suggestion.

4.3.3.2.2 Learning-based driver error modeling

We invite 17 volunteers with real-world driving experience to participate in this human-in-the-loop ramp merging simulation, as in Fig. 4-48. The drivers are guided to try their best to follow ASA during the simulation, so that the ego vehicle can perform the cooperative merging maneuvers in a smoother way compared to the scenario when no ASA is provided.

Fig. 4-48 One of the cooperative merging scenario at on-ramp built in Unity.

To reduce any system biases in the simulation results, volunteers are chosen from various backgrounds: 1 senior driver (age > 50), 2 mid-age drivers (30 < age <= 50), and 14 young drivers (age <= 30); 15 male drivers and 2 female drivers.

At the very beginning, each volunteer drives the vehicle on the simulator multiple times to collect data for training. Note that two different merging scenarios are developed in Unity, where the driver is randomly asked to drive either the ramp vehicle or the mainline vehicle. Additionally, only one volunteer at a time is allowed to enter the room of the simulator. Therefore, the volunteer will not have any prior knowledge regarding the traffic scenario, so his/her driving behavior totally depends on how well he/she can track the HUD-based ASA.

4.3.3.2.2.1 Driver type clustering and classification

To cluster the test subjects into different types according to the similarity of their driver behavior, for the observation of speed, four variables are measured during each run.

- The variance of speed (σ_v) describes the stability of the driving.
- The mean error of speed (μ_{Δ_v}) is the average difference between the advisory speed and actual speed, which evaluates the execution ability of the driver. Also, it distinguishes the driver who is always slower than the advisory speed from the driver who always exceeds the advisory speed.
- The absolute mean error of speed ($|\mu_{\Delta_v}|$) avoids misclassifying the driver who has a small mean error of speed, but actually drives pretty aggressively.
- The variance of the speed error (σ_{Δ_v}) is the variance of the difference between the advisory speed and the actual speed, implying the stability of the driver's execution.

Similarly, five variables in the observation of acceleration are also measured, including:

- The variance of acceleration (σ_a).
- The mean error of acceleration (μ_{Δ_a}).
- The absolute mean error of acceleration ($|\mu_{\Delta_a}|$).
- The variance of the acceleration error (σ_{Δ_a}).
- The mean of acceleration (μ_a).

Since the number of driver type is not strictly defined in this study, an unsupervised learning approach is used to cluster the driver. The pseudocode of this HCA is stated as *Algorithm 1*. The Euclidean distance and Ward linkage method, which are both HCA methods, are adopted to create a hierarchical cluster tree for clustering.

We combine each driver's data as a matrix X, X= $\{X_1,..., X_i,..., X_n\}$, where X_i = $\{\sigma_v$, μ_{Δ_v}, $|\mu_{\Delta_v}|$, σ_{δ_v}, μ_{Δ_a}, μ_a, $|\mu_{\Delta_a}|$, σ_a, $\sigma_{\Delta_a}\}$, and compute the Euclidean distance matrix D as

$$D = \begin{bmatrix} 0 & D_{12} & \cdots & D_{1n-1} & D_{1n} \\ D_{21} & 0 & \dots & D_{2n-1} & D_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ D_{n-11} & D_{n-12} & \dots & 0 & D_{n-1n} \\ D_{n1} & D_{n2} & \cdots & D_{nn-1} & 0 \end{bmatrix}$$

45

where $D_{ij} = \|X_i - X_j\|_2^2$.

Algorithm 4 HCA: Cluster the Driving Type.

Input: Matrix (X) that contains 9 variables of driver behavior and n samples.

Output: K clusters.

1: Compute the distance matrix.
2: **While** *the number of clusters > 1*
3: | Merge two clusters with the smallest D_{ij};
4: | Update the distance matrix;
5: | Save D_{ij} and cluster ID in the stack;
6: | The number of clusters is cut in half;
7: **end**
8: Separate from one cluster into several clusters based on the median distance among recorded D_{ij};

As explained in Algorithm 4, when the dendrogram is generated based on the similarity of each data sample, we cut the dendrogram by the median Euclidean distances among the samples to obtain the final clusters. After filtering out the outlier samples, all valid samples in the driver dataset are clustered into four major types. We adopt the Multidimensional Scaling (MDS) to display the driver type clustering result in 3-D space, which is shown in Fig. 4-49. Note that if more data samples are obtained, we might cluster them into more than these four types to achieve better performance.

Once the clustering is finished, we need to explore useful features from the data to obtain a more precise model. For instance, to compensate for the human tracking errors, we identify the most contributing variables as the object for the neural network to predict. Moreover, in high dimensions, there is little difference between the nearest and the farthest neighbor for the k-nearest neighbors (k-NN) classification using Euclidean distance because of "the curse of dimensionality", so we reduce the input variables for classification. As stated in Algorithm 5, PCA is used to transform these nine correlated variables into a

set of linearly uncorrelated variables, which are called principal components. Note PCA is not utilized before the driver clustering since the computational burden is not bottlenecked by the clustering, and all the original variables are potentially helpful for the clustering.

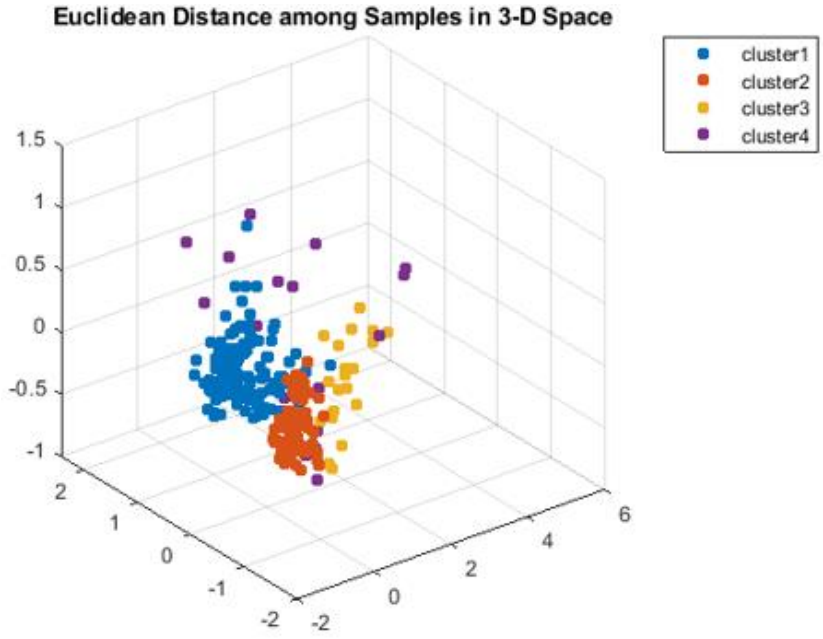


Fig. 4-49 HCA cluster visualization in 3-D space

As stated in the $n \times 9$ matrix of driver's data, we have nine types of features and n data samples. Specifically, we propose Algorithm 2 to identify the important variables to predict the speed tracking errors. According to the analytical results of PCA, the first component solely contributes 74.76%, and the second one solely contributes 18.85%. Several criteria for deciding how many components should be chosen are given in [167]: (a) the "elbow" in visual interpretation plot, (b) meaningful percentage of variance (80-90%), and (c) interpretable components. To meet these three criteria, we keep the first two components.

Algorithm 5 PCA:: Identify the Important Variables.

Input: Matrix X that contains nine variables of driver behavior.

Output: 1) Accumulate percentage of singular value (POS). 2) Correlation coefficients matrix.

1: Normalize the data matrix;
2: For each column X_j in X, $X_i = X_i - \mu$;
3: Calculate the covariance matrix
$K_{xx} = COV[X, X] = E[(X - \mu_x)(X - \mu_x)^T]$;
4: Calculate the singular values Σ and singular vector V, based on $K_{xx} = V\Sigma^2 V^T$;
5: Arrange Σ in descending order POS $= \frac{\Sigma_i}{sum(\Sigma)}$;
6: Calculate the correlation matrix (factor loading) R;

According to the correlation results, the variance of the speed errors (σ_{Δ_v}) has a good correlation with the first principal component, where $\sigma_{\Delta v}$ ranks the highest in the correlation, and the variance of speed (σ_v) stands out among the others. Having a higher correlation with the first two principal components, speed-related variables contribute much more than the acceleration on the driving behavior, so predicting and compensating for the speed errors can have a significant improvement in the execution of the advisory speed. We also notice the variance of the acceleration errors (σ_{Δ_a}) and the variance of acceleration (σ_a) play important roles, which can be considered as another two variables in the classification.

Once we obtain these different clusters by HCA and find out the important features by PCA, we can classify drivers into those clusters based on their driving behaviors during the time horizon of $t_{classify}$. By proposing the k-NN algorithm (as stated in Algorithm 6), we classify the driver into the same type as those that share the similar driving behavior.

Algorithm 6 ***k*-NN**: Classify the New Driver.

Input: 1) The speed trajectory of the driver during the time horizon $t_{classify}$. 2) Sample data of clustered drivers. 3) The number of neighbors to be considered.

Output: The type of the user.

1: Compute the data matrix X which contains the nine variables, where $X_i = \{\sigma_v, \sigma_{\Delta v}, \sigma_a, \sigma_{\Delta a}\}$;
2: Compute the similarity between the driver and all other previous drivers, where $S_i = \|X - X_i\|_2^2$;
3: Rank S and pick out the top k samples;
4: Do majority-voting
    For k samples
        If (sample = type1): {type1.VOTE +1}
        Else: {type2.VOTE +1}
    End
    If (type1.VOTE >= type2.VOTE): {Type = Type 1}
    Else: Type = Type 2

4.3.3.2.2.2 Training the nonlinear autoregressive (NAR) neural network

Once all historical data generated by various drivers are clustered, neural networks are trained to predict driver behavior. The speed errors generated from the driver when tracking ASA can be considered as a time series with high variations. Since it is generally difficult to model a time series using a linear model, we adopt the NAR neural network [168] in this study. It has been proved by Lapedes and Farber [169] that, time series can always be modeled by the following NAR model

$$\hat{y}(t + \delta t) = f\{y(t), y(t - \delta t), y(t - 2\delta t), \dots, y(t - (\tau - 1) \times \delta t)\} \quad (46)$$

where δt denotes a time step, and the speed tracking error y at time $t + \delta t$ is predicted using τ past values of the series. The structure of the NAR network can be seen as 0. To approximate the unknown function $f(\cdot)$, the neural network is trained by means of the optimization of network weights w's and neuron biases b's. The numbers of hidden layers

and neurons per layer are completely flexible, which can be optimized through a trial-and-error process. Note that more neurons may complicate the system, but less neurons may restrict the generalization capability and computing power of the network.

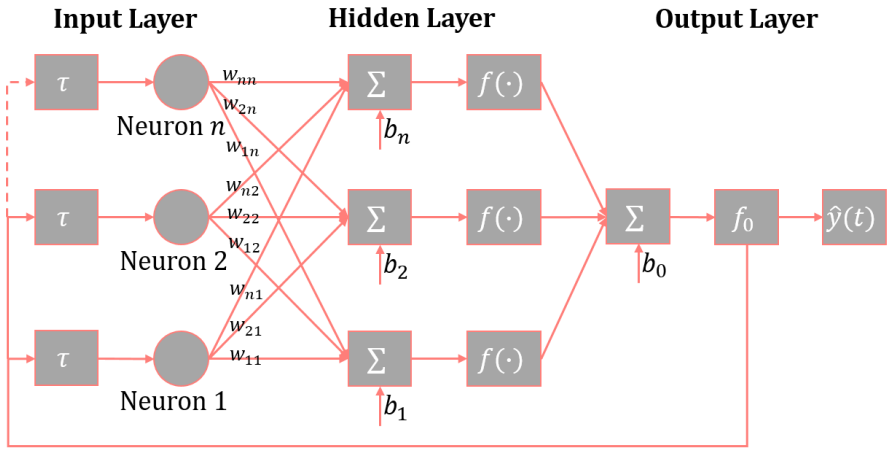


Fig. 4-50 Structure of NAR neural network

In this study, to train the series of speed error data, we set the number of hidden layers as 2, and the number of hidden neurons as 10. The number of delays τ is set to 2, which means a total of 3 values of the speed tracking errors are used to predict the value at the next time step. The Levenberg-Marquardt backpropagation procedure (LMBP) is implemented as the learning rule of this NAR network [170] LMBP is considered one of the fastest backpropagation-type algorithms, since it was designed to approximate the second-order derivative without computing the Hessian Matrix. The training process is conducted on the Windows desktop with processor Intel Core i7-7700K @ 4.20 GHz and 64.0 GB memory.

We evaluate our training result using the Mean Squared Error (MSE), which is the average squared difference between predictions and targets, and the Regression (R) value, which is a measurement of the correlation between output predictions and targets. For

MSE, a lower value stands for a better result where zero means no error. A higher R value means a stronger correlation between the prediction and target, while zero stands for a random relationship. We pick two biggest clusters out of those four shown in 0, and split 70% for training, 15% for validation, and another 15% for testing. The Table XIII shows example results for two driver cluster, where two neural networks are trained with high performance. These two networks' MSE values are lower than 0.02 and R values are higher than 0.99.

TABLE XIII TRAINING RESULT OF NAR NEURAL NETWORK

Data Set	Catalog	Target Values	MSE	R
Driver Type 1	Training	20512	0.0147	0.9906
	Validation	4396	0.0067	0.9959
	Testing	4396	0.0073	0.9956
Driver Type 2	Training	8330	0.0166	0.9929
	Validation	1785	0.0140	0.9939
	Testing	1785	0.0191	0.9953

Once the neural networks are completely trained, they can be implemented in an online manner as shown in Fig. 4-46. At every time step, the trained neural network (configured as a MATLAB script) takes multiple inputs through the UDP socket from either the game engine or the vehicle, computes the predicted speed error at the next time step, and sends it back through the UDP socket. Once the predicted speed error is received, it will then be compensated for the original advisory speed algorithm Equation 46 by

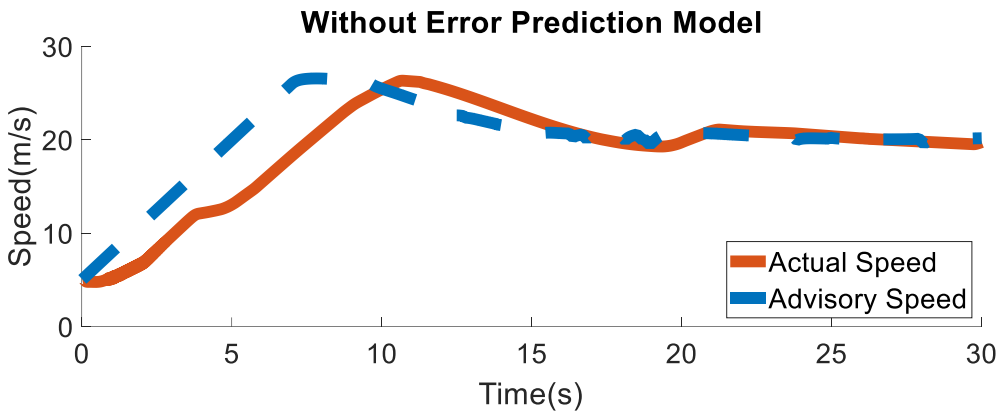
$$\hat{v}_i(t+\delta t) = v_i(t) + a_{ref}(t+\delta t)\cdot\delta t + \hat{y}(t+\delta t) \tag{47}$$

where $\hat{y}(t+\delta t)$ is the predicted error term compensated for the advisory speed. This compensated advisory speed $\hat{v}_i(t+\delta t)$ is the value that is eventually displayed to the driver.

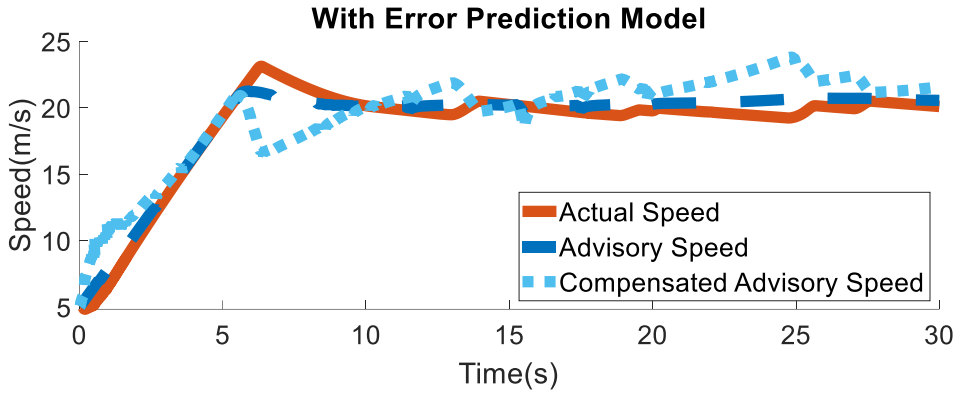
4.3.3.2.3 Result of human-in-the-loop simulation

As stated in the previous section, 17 volunteer drivers were invited to train the NAR neural network in the offline training phase, so the data they generate are considered as "historical data by various drivers" in Fig. 4-46. However, since the proposed driver behavior modeling methodology is for unknown drivers, we invite another five drivers to test the system in the online actuation phase. Each driver conducts eight simulation trips, so a total of 40 runs are recorded for evaluation.

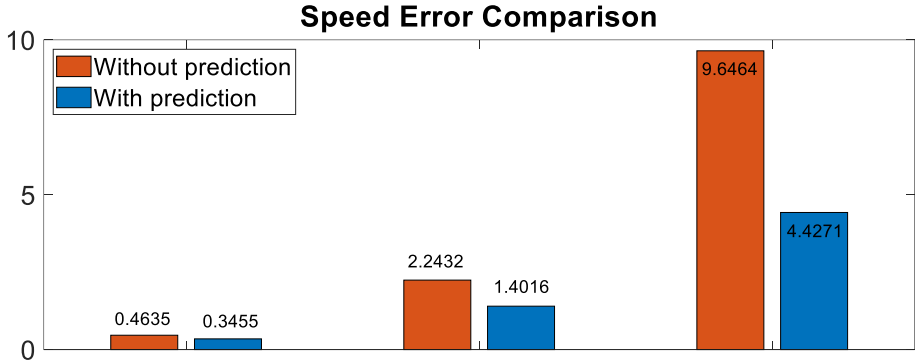
Two out of those 40 speed trajectories generated by human-in-the-loop simulation runs are selected to conduct an illustrative comparison in Fig. 4-51, and a better tracking of the advisory speed is observed after implementing the proposed model in general. Note for all speed trajectory figures in this study including Fig. 4-51 (a) (b): At the same time step, the advisory speed is first generated, the compensated advisory speed is the second (if there is one), and the actual speed is the last. They are not generated at the same time, as illustrated in the system workflow in Fig. 4-46.



(a)



(b)



(c)

Fig. 4-51 Speed error comparison in the game engine-based simulation

As shown in Fig. 4-51(a), there is a large speed difference (which indicates a poor speed tracking behavior) at the beginning of the 30-second simulation run, when there is no error prediction model. During this whole run, the red solid line and the dark blue dashed line

are not well aligned with each other, indicating that the actual speed generated by the driver deviates from the advisory speed generated all the time.

However, as shown in Fig. 4-51(b), the light blue dotted line denotes the compensated advisory speed calculated by Equation 47 which predicts driver behavior based on his/her previous driving inputs. For example, at time 7 s, the actual speed (23 m/s) is lower than the advisory speed (21 m/s), so the speed tracking error is -2 m/s. This value along with the values of two previous time steps are the time series inputs of the neural network. The neural network then outputs the prediction speed tracking error at the next time step, which is -3 m/s. This predicted error is added to the advisory speed at time 8 s, so the compensated advisory speed at 8 s is (20 – 3 =) 17 m/s. With the help of the compensated advisory speed, the speed errors are shown to be attenuated during 6-10 s.

In general, with the compensated advisory speed calculated by the error prediction model, the driver can track the advisory speed more precisely than without it, since the red line and the dark blue line are generally closer and less fluctuated in Fig. 4-51 (b) than Fig. 4-51 (a). As shown in Fig. 4-46, the compensated advisory speed is only adopted in the loop for display purpose, where the speed tracking errors are still calculated by the difference between the actual speed and the advisory speed.

As for the quantitative comparison, we evaluate three different indexes for all 40 runs (with and without the error prediction model) conducted by five drivers. Those three indexes include the mean speed error μ_{δ_v}, the mean value of the absolute speed error $|\mu_{\delta_v}|$, as well as the variance of the speed error σ_{δ_v}. As shown in Fig. 4-51 (c), the mean speed error benefits the least from implementing the error prediction model compared to the other

two indexes, with a 23.4% reduction of this index. However, if we take an absolute value of the speed error first, a 36.2% reduction of the index can be observed, which outperforms the previous one. The underlying reason is that the absolute calculation filters out the situations when the speed errors are bouncing up and down, and positive values offset negative values so the mean values turn out to be relatively small.

As a matter of fact, the speed error variance results in Fig. 4-51 (c) prove the effectiveness of the proposed driver behavior model in an even better way. As shown in the results, the speed error variance is 9.6464 before the error prediction model is implemented and is cut by half to 4.5661 after the implementation. This 52.7% drop in speed error variance shows that drivers are capable of tracking the advisory speed more closely after the driver behavior model is implemented.

Additionally, we also utilize the U.S. Environmental Protection Agency's MOtor Vehicle Emission Simulator (MOVES) model to perform analysis on the environmental impacts of the proposed model based on all human-in-the-loop simulation runs [158]. As can be seen from Table XIV, the pollutant emissions can be reduced by up to 6.3% after implementing the driver behavior model, and the energy consumption can be reduced by 2.5%, respectively.

TABLE XIV ENERGY CONSUMPTION AND POLLUTANT EMISSION RESULTS OF HUMAN-IN-THE-LOOP SIMULATION (ALL VALUES ARE ON A KILOMETER BASIS)

	CO (g)	HC (g)	NOX (g)	CO2 (g)	Energy (KJ)
Baseline	2.54	0.0175	0.05721	275.1	3867
Proposed	2.43	0.0167	0.05359	268.2	3770
Reduction	4.3%	4.6%	6.3%	2.5%	2.5%

4.3.3.2.4 Cooperative ramp merging based on ASA: system design and field implement

After the successful simulation and evaluation of the Advisory Speed Assistance (ASA) system in the HuiL co-simulation platform, the next crucial step is to move towards real-world implementation. Transitioning from simulations to field implementation enables us to assess the system's performance under real driving conditions and evaluate its practicality and effectiveness. In this section, we present the details of the field implementation process and describe the vehicle-edge-cloud test bed that was utilized for deploying the ASA system. Through this field implementation, we aim to gather real-world data, validate the system's performance, and gain insights into the challenges and opportunities that arise in a live driving environment.

In the previous section, we develop a method to provide accurate speed suggestion to connected vehicle drivers for coordinating the ramp merging in simulation, but to implement the cooperative merging and speed error compensation algorithms in real-world, there are still two major issues waiting to be addressed.

- The major method, Dedicated Short-Range Communications (DSRC), is not a suitable method for the scenario where different vehicles are not on the same altitude, or there are major obstructions between two DSRC On-Board Units (OBU).
- Driving behavior modeling needs a large amount of data, and online behavior analysis requires real-time update from vehicles.

Therefore, vehicle-to-cloud (V2C) communication comes to our sight. Compared with DSRC method, the V2C method improves the scalability of vehicle communication, provides a larger data storage which enables data driven human behavior modeling, and

breaks the constraint of computational power. In this study, we adopt 4G/LTE-based vehicle-to-cloud communication to design our cooperative ramp merging system, supported by the Vehicle-Edge-Cloud (VEC) real-world implementation platform.

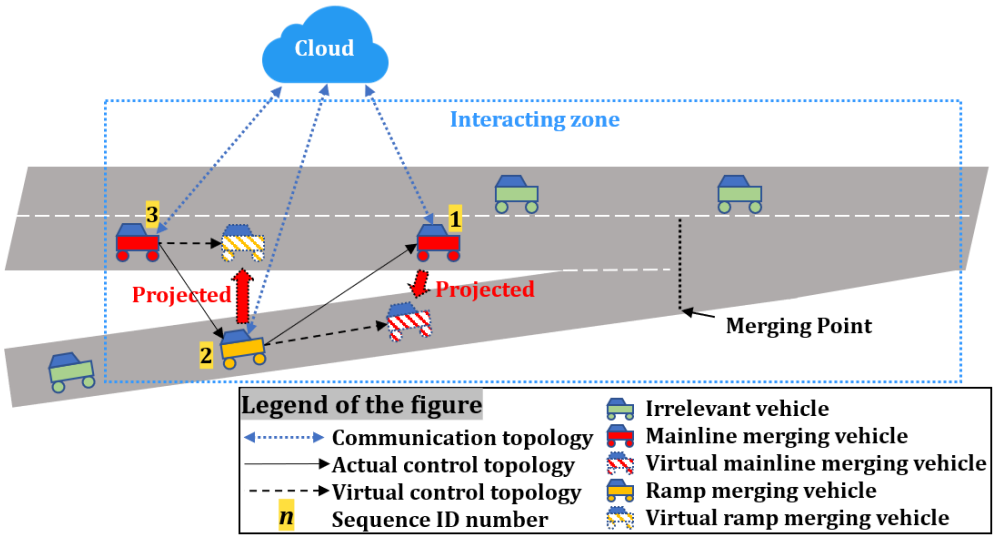


Fig. 4-52 V2C-based cooperative merging scenario at on-ramp

4.3.3.2.4.1 Real world ASA implementation specifications

The objective of this study is to design a V2C-based ramp merging coordination system using the aforementioned ASA, so drivers can drive those vehicles with the help of this ASA system to conduct safe and smooth merging maneuvers with other merging vehicles cooperatively. The proposed system can be simply illustrated as Fig. 4-52, where all relevant merging vehicles are assumed to have access to the server through V2C communication. The server processes the data received from various vehicles in real time, and also sends advisory information back to the relevant vehicles with certain updating frequency.

This implementation mainly utilizes the edge server layer and vehicle layer of the VEC platform (Fig. 4-1). As a realization of the “Vehicle Digital Twin” concept, a flexible

cloud-based CAV system framework has been developed and demonstrated. Real-world field implementation of the proposed ramp merging system has been conducted with three passenger vehicles. As the prototype of the simulation road network, the test track consists of a ramp and a mainline, where the mainline spans from the intersection of Columbia Avenue and Chicago Avenue to the intersection of Iowa Avenue in Riverside, California. In the Vehicle Digital Twin framework presented in Fig. 4-53, onboard devices upload the data to the cloud server through the 4G/LTE cellular network. The server creates Digital Twins of vehicles and drivers whose parameters are synchronized in real-time with their counterparts in the physical world, processes the data with the proposed models in the digital world, and sends advisory information back to the vehicles and drivers in the physical world. Fig. 4-53 shows how the speed guidance is shown in a uniquely built user interface, where the key modules on the server are introduced.

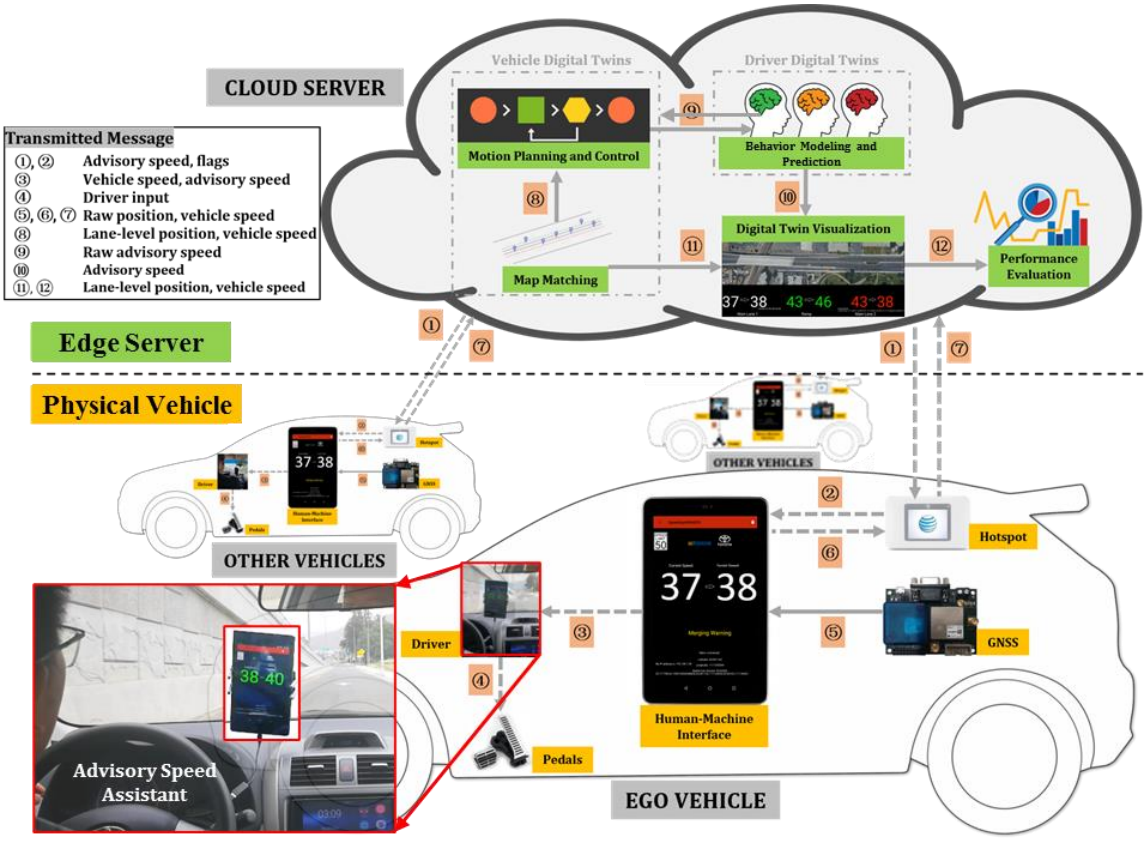


Fig. 4-53 General architecture of the real-world cooperative ramp merging ASA system

There are four main modules on the server,

1) Map matching module:

The main functions of the map matching module are position synchronization and geo-fencing. A pre-built map of the test field is available on the cloud server, with information such as the road type, road length, road ID, waypoints, direction, road speed limit, merging zone, and interacting zone. For position synchronization, vehicles' coordinates (i.e., longitude, latitude, and altitude) received from the GNSS will be uploaded to the cloud at each time step and be matched to the pre-built map by the proposed map matching algorithm (Based on the position given by the map matching algorithm, a geo-fencing

function is defined to check the positions and conditions of the vehicles at each time step, so associated actions can be conducted accordingly.

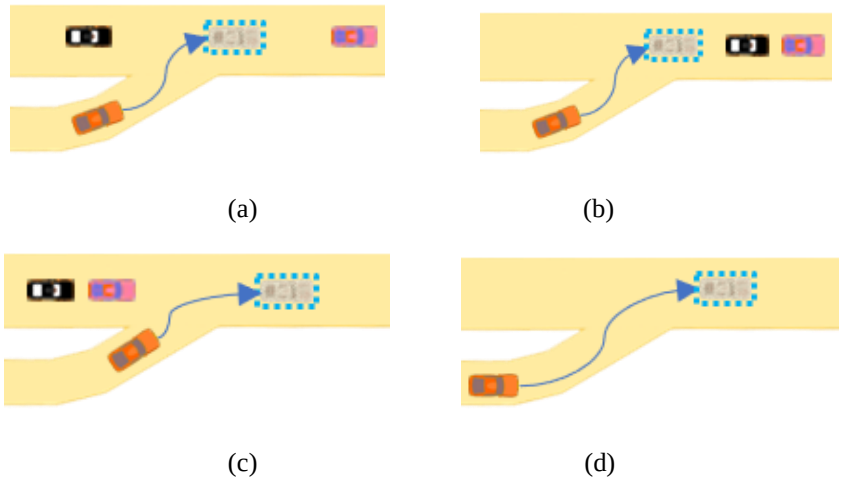


Fig. 4-54 Merging scenarios: (a) merging into a vehicle string, (b) merging after a vehicle or string, (c) merging in front of a vehicle or string, and (d) merging to an empty mainline

2) Motion planning module:

Motion planning module is designed to cooperatively plan the motions of three merging vehicles, with two vehicles on mainline and one on ramp. In each merging, module will recognize the scenario and assign each vehicle its role. Considering the succinctness of description, with regard to the initial state, we name the first vehicle on mainline as "MV1", the second vehicle on mainline as "MV2", and the vehicle on ramp as "RV", hereafter. Shown as Fig. 4-54, there are different ramp merging scenarios, including 1) merging into a vehicle string (of two or more vehicles), 2) merging after a vehicle or string, 3) merging in front of a vehicle or string, and 4) merging to an empty mainline. In Fig. 4-54(a), merging to a busy mainline and cutting into a string is the most challenging case for human drivers among these scenarios. Therefore, the following analysis will be based on this representative case, with two mainline vehicles and one ramp vehicle. Note that the overall

system designed in this study can handle all these four cases by the game theory-based merging sequence determination algorithm [145].

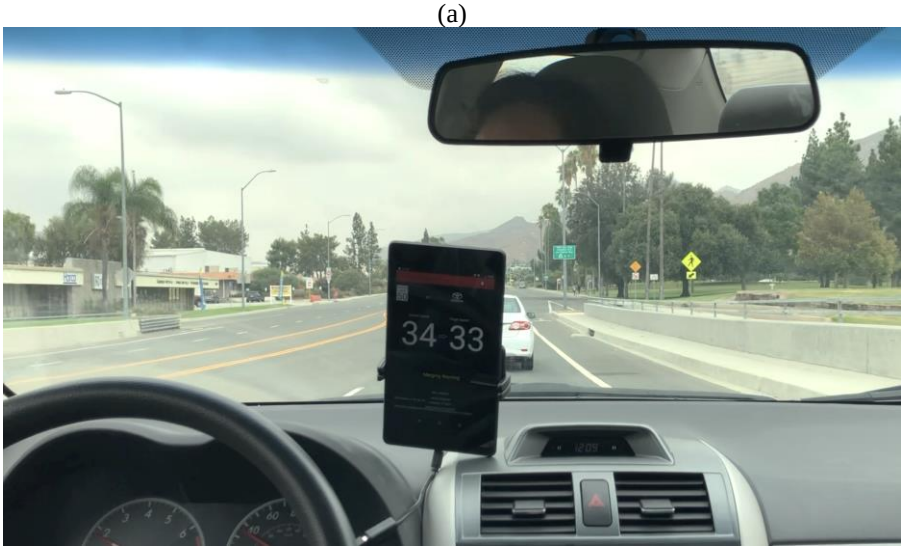
3) Control module:

The motion controller on the cloud server generates the raw advisory speed of the ego vehicle. The inputs of the controller are the speed and lane-level position of the ego vehicle and its preceding (virtual) vehicle, where the preceding vehicle is decided by the motion planner in our previous research [145].

4) Human behavior module:

Given the fact that a driver cannot track the advisory speed shown on the HMI perfectly, a speed tracking error is always compensated at every time step, as introduced in the last Section 4.3.3.2.2. As shown in Fig. 4-55, the information displayed on the HMI include current speed (the left number), advisory speed (the right number), speed limit, and some other additional messages (e.g., latitude and longitude, IP address). Besides visual information, audio notification is also embedded in this HMI. For example, along with showing the yellow text in Fig. 4-55(b), an audio warning is played to driver of MV2 when the ramp vehicle is approaching.

(a)



(b)

Fig. 4-55 Perspectives of (a) ramp vehicle driver and (b) mainline vehicle 2 driver while conducting the cooperative ramp merging

The test track consists of a ramp and a mainline, where the mainline spans from the intersection of Columbia Avenue and Chicago Avenue to the intersection of Iowa Avenue in Riverside, California. The mainline is on an overpass while the ramp is under the overpass, which increases the difficulty of ramp merging because the visions of both

mainline and ramp drivers are blocked by this overpass. The total length of the track is 780 meters, with a merging zone of 89 meters long.

(a)

(b)

(c)

(d)

Fig. 4-56 V2C-based cooperative merging stages

As shown in Fig. 4-57, the cooperative merging consists of four stages. On stage 1 in Fig. 4-57(a), MV2 is assigned to follow MV1 and enters the interacting zone with a constant speed. At the same time, RV receives the countdown information from the approaching MV1. On stage 2 in Fig. 4-57(b), RV is assigned to follow MV1 and starts to accelerate based on the speed suggestion. On stage 3 in Fig. 4-57(c), MV2 is assigned to follow RV when RV satisfies the requirement. RV is ready to merge, while MV2 is notified to slow down and generates a gap for the merge. On stage 4 in Fig. 4-57(d), given enough inter-vehicle gap, RV merges into the vehicle string. The implementation setups are listed in TABLE I, including the preset initial states, constraints, desired states, and the parameters in control algorithm. The control gain of each vehicle will be changed at certain time steps as the merging process goes along. For RV, K_r is given by the feedforward lookup table based on the initial condition when RV and MV1 are hooked up. For MV2, K_{m1} depends on the initial condition when MV1 and MV2 are bonded; K_{m2} depends on the initial condition when MV2 switches its leader from MV1 to RV.

At every time step, the localization coordinates and speed provided by the GNSS units will be read by the HMI device on each vehicle and transmitted to the cloud server. On the cloud, the map matching module utilizes the GNSS data to locate these three vehicles and compute the relative distance and speed among them. With the map matching result, the real-time advisory speed for each vehicle can be calculated. More specifically, to smooth the speed and facilitate the following of traffic rules in the test site, an internal acceleration limit for the ramp vehicle is set in the control algorithm with a range of $(-3, 3)\ \mathrm{m/s^2}$, and the speed display on HMI is bounded by the real-world's speed limit 45 mph

(approximately 20 m/s). It should be noted that, when the advisory speed given by the consensus algorithm exceeds the speed limit, it will not be displayed on the HMI, but only increases the possibility of the HMI deactivation in the background.

4.3.3.3 Result Evaluation

The results evaluation in this section is conducted by the "Performance Evaluation" module, which is executed on the cloud server as shown in Fig. 4-53 in real time. With a certain update frequency, this module evaluates the speed and distance trajectories, as well as the fuel consumption and pollutant emissions of all connected vehicles. The evaluation results are sent to the drivers to provide them with real-time feedback, and therefore recommend safer or more fuel-efficient driving behaviors. Show as Fig. 4-57, an app is designed to visualize vehicle Digital Twins running on the cloud server, which have the same parameters (position, speed, etc.) as the vehicles in the real world. Speed advisories for drivers of all three vehicles are also updated in real time. In field experiments, two inspectors, one in the passenger seat, another one stay beside the server PC, monitor the whole process using this app.

A baseline implementation is designed and carried out to validate the benefit of the proposed cooperative merging. Since an unbiased comparison requires the same merging scenario, a low-grade HMI with only partial function is adopted in the baseline scenario. To ensure the three vehicles encounter each other, the HMI provides speed guidance to RV only before RV reaches a "deactivate" point. The point is set at 85 meters before the observable point, where the driver of RV drives "blindly" (without HMI or line-of-sight of

other merging vehicles) within that 85 meter range. Without speed guidance, MV2 will not create gap for RV in advance.

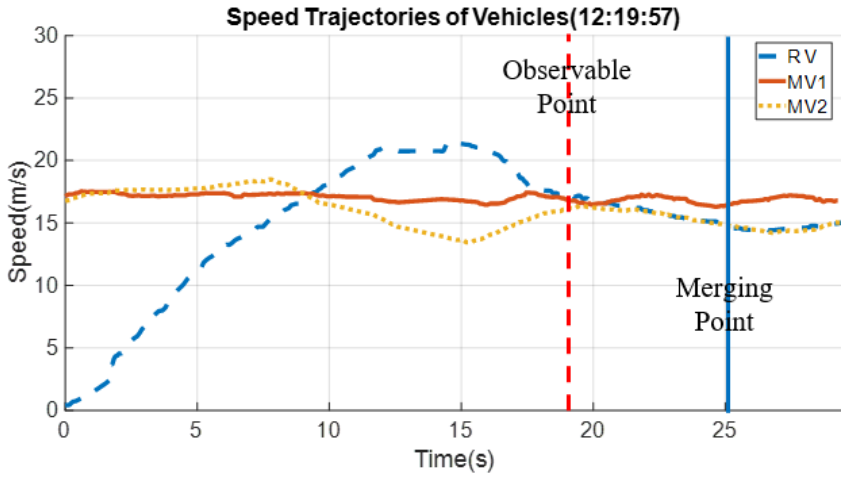
In our field implementation, four cooperative merging trips and four baseline trips are carried out, respectively. In this section, we present one typical cooperative merging trip that was carried out at 12:19:57 p.m. on September 27th, 2019. The proposed vehicle-to-cloud based cooperative merging system is compared with the baseline system in terms of safety, fuel and emission. Moreover, an implementation is conducted to investigate V2C communication.

Fig. 4-57 Digital Twin of vehicles running on the cloud server in real time

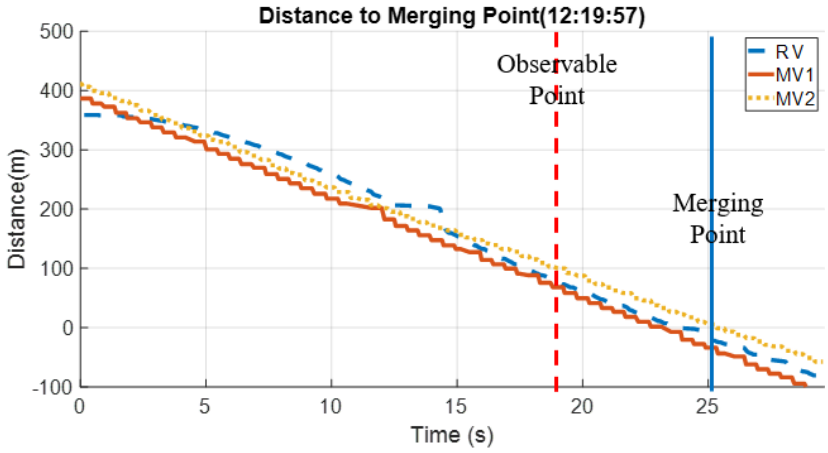
4.3.3.3.1 Safety results

As can be seen from the cooperative merging implementation results, Fig. 4-58(a) shows the speed trajectories, and Fig. 4-58 (b) shows the distance to merge of the three vehicles. The observable point is 100 meters before the merging point, where mainline and ramp drivers can see each other for the first time during merging. In this trip, drivers rely more on the HMI for longitudinal speed control because they have no visual information about the merging condition. After passing the observable point, when they have line-of-sight to

other merging vehicles, they would usually trust more in what they see and make their own decision, instead of relying on the HMI.



(a)



(b)

Fig. 4-58 Cloud-based cooperative merging scenario at on-ramp

During 0 – 8 seconds in Fig. 4-58, RV accelerates to close its gap with the string on the mainline, and MV1 and MV2 keep a relative constant speed. During this period, both RV and MV2 consider MV1 as their leader. At 8 seconds, MV2 switches its leader from MV1 to RV and decelerates. During 8 – 15 seconds, the gap between MV1 and MV2 has been

generated for RV to merge in. During 15 – 20 seconds, the speed of RV and MV2 converge to the speed of MV1, which means the cooperative longitudinal speed adjustment of three vehicles has already been completed before they actually conduct the lane change behaviors.

There are many approaches to measuring driving safety, such as time-to-collision, time-exposed time-to-collision, time integrated time-to-collision, crash index, headway, time-to-accident, and post-encroachment time to name a few. However, these methods are not applicable in this implementation because they require all the vehicles to be driven on the same lane with potential conflicts. In this implementation, MV1 is not affected by the merging behavior that happened behind, while MV2 and RV are initially on different lanes without direct conflict. Thus, we adopt speed variance as an alternative safety measurement method. Since the accidence rates increase with increased speed variance for all classes of roads, a safe merging environment can be created by reducing the speed variance. The speed variance of MV2 between the "deactivate" point and the merging point is an indicator of how upstream traffic safety is affected by the merging behavior.

During the normal merging process, both mainline drivers and ramp driver have to modify their speed, hence the speed variance of the involved vehicles will increase. The average speed variances of MV2 of during four baseline trips is 2.163, while the value is decreased to 0.705 in the four cooperative merging trips. The results show a reduction of 67.41% in terms of average speed variance, proving that the cooperative merging approach is safer than the baseline scenario.

4.3.3.3.2 Fuel and emissions results

For every trip, the fuel and pollutant emissions are analyzed by this Performance Evaluation module on the cloud server with the open-source "MOVESTAR" fuel and emission model [158]. To minimize the biases, MV1 is excluded in this estimation since it tracks a constant speed during the whole merging process and has no significant impacts.

The average fuel consumption and pollutant emissions of involved vehicles are calculated for both baseline and the proposed cooperative merging method, which is shown in Table XV. A reduction up to 31.21% in pollutant emissions and a reduction of 7.45% in fuel consumption can be obtained after implementing the proposed model comparing to the baseline, respectively.

TABLE XV ENERGY CONSUMPTION AND POLLUTANT EMISSION RESULTS

	CO (g/km)	HC (g/km)	NO_X (g/km)	CO_2 (g/km)	Fuel (g/km)
Baseline	0.88	0.0088	0.031	242.8	76.038
Proposed	0.61	0.0056	0.025	224.7	70.375
Reduction	31.21%	35.71%	20.00%	7.46%	7.45%

4.3.3.3.3 Communication results

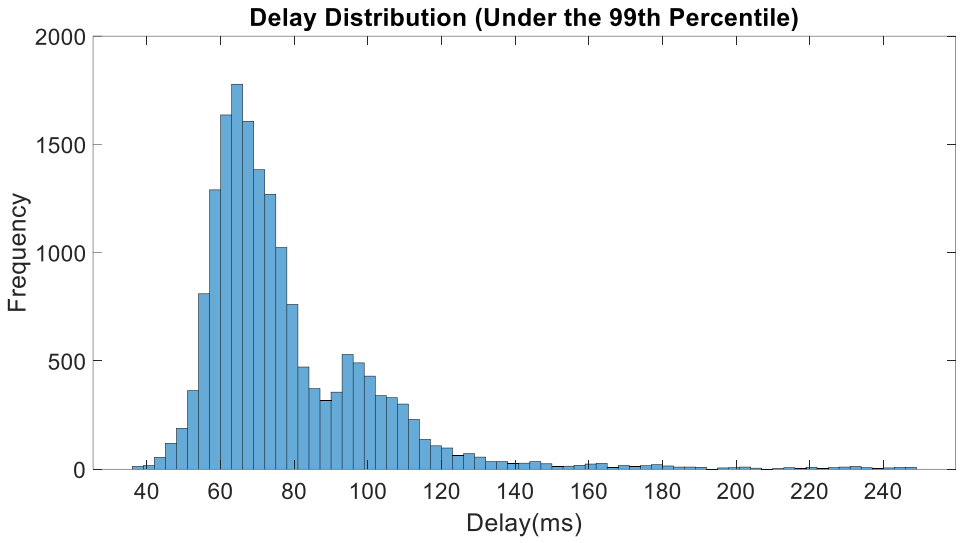
A comprehensive test is conducted to measure the delay of 4G/LTE-based V2C communication. In the test, in order to measure only the communication time, all the algorithms are wiped out on the server with only the communication module remained to eliminate the computation time.

The communication delay is defined by a back-and-forth time differences of a "handshake" message. First, the ego vehicle sends the message to the target vehicle and records the current time t_1. Since the target vehicle is designed to reply immediately, when

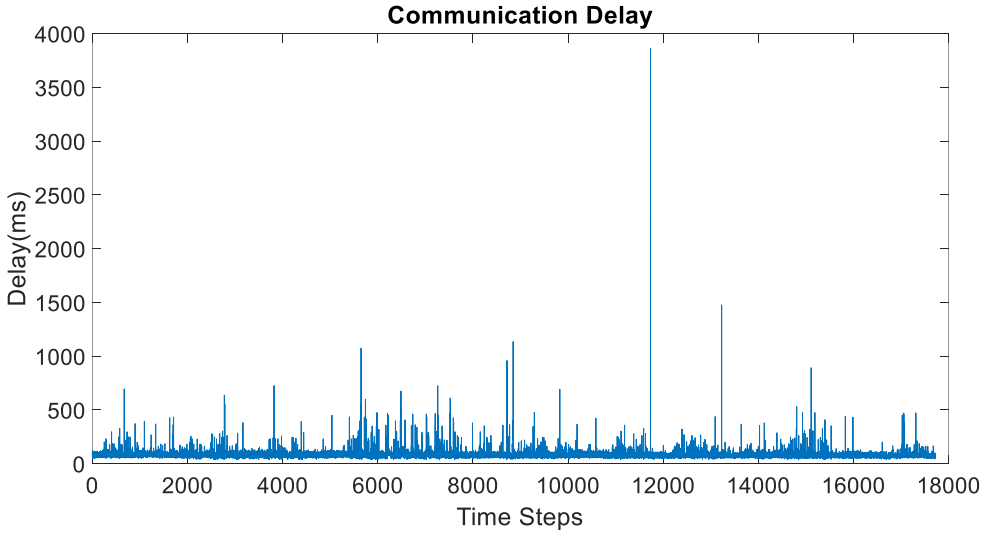
the ego vehicle receives the reply message and at time t_2, the communication delay is calculated as $t_2 - t_1$.

A total of 17717 communication delay samples are recorded during the field implementation from five different trips. Fig. 4-59(a) shows the distribution of communication delay under its 99th percentile. The majority of the value (65%) is located in the range of 54-81 ms, where this 27 ms span only takes up 12.6% of the whole span. Shown in Table XVI, the 99th percentile of communication delay is 247 ms, which is smaller than the update frequency of the HMI (3 Hz, 333 ms). Since the computation of the algorithm takes only 30 ms, 99% of the V2C communication is quicker than HMI update timeframe and will not undermine the performance of the algorithm.

According to Fig. 4-59(b), delay surges appear occasionally, and this 1% of high delay cannot be neglected since the highest delay reaches 3861 ms. Bringing in a Kalman filter or simply keeping the original speed suggestion are the solutions to handle this problem.



(a)



(b)

Fig. 4-59 Communication delay results: (a) Time series of communication delay and (b) Distribution of communication delay under its 99th percentiles

TABLE XVI STATISTIC RESULT OF COMMUNICATION DELAY

Average	Minimum	Maximum	75th Percentile	99th Percentile
80 ms	36 ms	3861 ms	88 ms	247 ms

Additionally, we also intend to conduct a comparison test regarding the communication methods, i.e., 4G/LTE-based V2C communication versus DSRC-based V2V communication. We install two DSRC OBUs (Savari MobiWAVE 1000) on two of the vehicles (i.e., MV1 and RV), and allow them to send information to each other while the vehicles are conducting the cooperative merging. However, during the merging process, there is a major period of time (more than 5 seconds) that the information cannot be successfully transmitted between these two OBUs. That period mostly happens when MV1 is on the top of the overpass while RV is on the ground (which can be roughly shown as Fig. 4-56(a)). As a similar phenomenon is reported by [171], a box trailer will likely "block" the DSRC signal broadcast since it requires line-of-sight for reception. Such performance shows that DSRC-based V2V might not be a good communication method for scenarios like this, when different vehicles are not on the same altitude, or there are major obstructions between two OBUs.

DSRC-based V2I communication might be a good alternative to solve the non-line-of-sight issue of V2V communication, but V2I communication with roadside units also have disadvantages of transmission range, computation power, and data storage compared to V2C communication with cloud computing. In this study, computation power and data storage are essential to conduct human behavior modeling.

4.3.3.4 Conclusions and Future Work

In this study, to extend the algorithm of intelligent vehicles to human driven vehicles, a cooperative ramp merging ASA system has been developed using a V2C Digital Twin approach. Field implementation in the real world has been conducted with three typical passenger vehicles, which shows the proposed system improves the current ramp merging scenario in terms of safety and environmental sustainability. Specifically, compared with the baseline scenario with no advisory information during the merging process, the proposed system reduces the average speed variance by 67.41%, reduces the pollutant emissions by up to 31.21%, and reduces the fuel consumption by 7.45%, respectively.

As the first few real-world implementations of Digital Twin in the automotive domain, numerous research questions need to be solved along its future development. To name a few, what is the required update frequency of the Digital Twins on the cloud server, so it can tolerate the uncertainties brought by V2C communication delay and packet loss. Additionally, what other modules can be designed and placed on the cloud server, so the functionality of the Digital Twin architecture can be fully utilized. Implementing the proposed V2C Digital Twin approach to other traffic scenarios besides cooperative ramp merging is also one of our future studies.

5 Conclusions and Future Work

5.1 Conclusions

In the concluding remarks of this dissertation, we reflect on the significant strides made in the exploration and implementation of personalized behavior-aware motion planning framework of Intelligent Vehicles. In this dissertation, three main contributions are made, including the construction of platforms, driver behavior research, and the development of behavior-aware motion planning algorithms.

- We have successfully created a vehicle-edge-cloud digital twin platform for real-world implementation and a human-in-the-loop co-simulation platform, which constituted the first step towards understanding personalized driving behavior. These platforms, working in conjunction, enable the generation of driving behavior datasets and facilitate the development and validation of advanced algorithms. Such an integrative and systematic approach has notably improved the practical realization of intelligent vehicle applications.
- Our study further formulated a systematic framework to investigate personalized driving behavior. This included an examination of various aspects such as driver profiling, driving preferences, and implicit interaction patterns. Through this framework, we were able to model unique driving behavior and develop individualized models that would feed into the intelligent vehicle systems. This integrative approach provided a more nuanced and accurate understanding of driving behavior, enabling the design of more effective and favorable autonomous driving systems. As a result, the system can recognize the target driver's lane

change intention at 6.08 seconds on average before the lane change, and the predicted trajectory is 1.03 m within a 4-second prediction window. Using a personalized model can improve the prediction accuracy by 27.8%, as in Section 4.3.1.

- Finally, we proposed a behavior-aware motion planning strategy for intelligent vehicles operating in mixed traffic environments. This strategy brought several significant benefits to the transportation system, including improved safety (reduces the average speed variance of the traffic by 67.41%, as in Section 4.3.3), efficiency (improve system mobility by up to 210%, as in Section 4.3.2), environmental sustainability (reduce fuel consumption by up to 53.9% and pollutant emissions by up to 31.21%, as in Section 4.3.3) and user experience (eliminated the extreme speed values, as in Section 4.3.2). The strategic integration of behavioral awareness in motion planning enabled vehicles to respond more adaptively to dynamic traffic conditions, enhancing the overall functionality of the transportation system. We also extended the algorithm for Intelligent Vehicles to benefit human-driven vehicles, utilizing simple, portable, and easily installable hardware. This extension ensures that the benefits of our research reach even beyond IVs, fostering an inclusive and optimized transportation system for all.

In conclusion, our research has laid the groundwork for an in-depth understanding of personalized driving behavior and its application to intelligent vehicles. We anticipate that the insight from this work will continue to facilitate the development of IV technology and contribute to a safer, more efficient, and user-centric eco-friendly transportation system.

5.2 Related Publications to This Dissertation

[1] X. Liao, X. Zhao, Z. Wang, Z. Zhao, K. Han, R. Gupta, M. J. Barth, and G. Wu, "Driver digital twin for online prediction of personalized lane change behavior," *IEEE Internet of Things Journal*, 2023.

[2] X. Liao, G. Wu, M. J. Barth, R. Gupta, and K. Han, "Exploring vehicular interaction from trajectories based on granger causality," *IEEE Intelligent Vehicles Symposium*, 2023.

[3] X. Liao, Z. Wang, K. Han, X. Zhao, M. Barth, and G. Wu, 15 Driver Behavior-Aware Cooperative Ramp Merging for Intelligent Vehicles. *Walter de Gruyter GmbH & Co KG*, 2023, vol. 3.

[4] X. Liao, G. Wu, L. Yang, and M. J. Barth, "A real-world data-driven approach for estimating environmental impacts of traffic accidents," *Transportation Research Part D: Transport and Environment*, vol. 117, p. 103664, 2023.

[5] X. Liao, Z. Wang, X. Zhao, Z. Zhao, K. Han, P. Tiwari, M. J. Barth, and G. Wu, "Online prediction of lane change with a hierarchical learning-based approach," in *IEEE International Conference on Robotics and Automation (ICRA). IEEE*, 2022, pp. 948–954.

[6] X. Liao, S. Mehrotra, S. Ho, Y. Gorospe, X. Wu, and T. Mistu, "Driver profile modeling based on driving style, personality traits, and mood states," in *IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 709–716.

[7] X. Liao, X. Zhao, Z. Wang, K. Han, P. Tiwari, M. J. Barth, and G. Wu, "Game theory-based ramp merging for mixed traffic with unity-sumo co-simulation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 52, no. 9, pp. 5746–5757, 2021.

[8] X. Liao, Z. Wang, X. Zhao, K. Han, P. Tiwari, M. J. Barth, and G. Wu, "Cooperative ramp merging design and field implementation: A digital twin approach based on vehicle-to-cloud Communication," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 5, pp. 4490–4500, 2021.

[9] Z. Wang, X. Liao, X. Zhao, K. Han, P. Tiwari, M. J. Barth, and G. Wu, "A digital twin paradigm: Vehicle-to-cloud based advanced driver assistance systems," in *IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–6.

[10] Z. Wang, X. Liao, C. Wang, D. Oswald, G. Wu, K. Boriboonsomsin, M. J. Barth, K. Han, B. Kim, and P. Tiwari, "Driver behavior modeling using game engine and real vehicle: A learning-based approach," *IEEE Transactions on Intelligent Vehicles,* vol. 5, no. 4, pp. 738–749, 2020.

[11] X. Zhao, X. Liao, Z. Wang, G. Wu, M. Barth, K. Han, and P. Tiwari, "Co-simulation platform for modeling and evaluating connected and automated vehicles and human behavior in mixed traffic," *SAE International Journal of Connected and Automated Vehicles,* vol. 5, no. 4, 2022.

5.3 Future Work and Discussion

Moving forward, there are still several intriguing paths for further exploration and research in the field of intelligent vehicles.

- First, the concept of a human-centric system, rooted in behavioral modeling, presents a compelling area for potential investigation. The cornerstone of this approach would be the development of optimal and favorable automation technologies that are explicitly designed to align with drivers' preferences and abilities. To further improve the human-centric system, the modeling of relationships and interactions in multi-agent and human-vehicle contexts could prove invaluable. Additionally, a key focus should be placed on enhancing the acceptance and trust in automation technologies by developing explainable methods.
- In addition to refining individual vehicle behaviors, we should also consider extending our planning strategies to encompass a broader traffic perspective. This includes improving the scalability of our algorithm to optimize traffic flow and creating more robust and efficient transportation systems.
- Lastly, it is crucial to address the gap that exists between academic research and practical implementation. By collecting more comprehensive and naturalistic personalized driving datasets, we can develop models that better reflect real-world driving behavior. Further, considering engineering problems associated with the Digital Twin framework can bring us a step closer to seamless and efficient implementation of intelligent vehicle technologies, such as addressing the computational demand allocation, communication latency, driver model update, etc.

Along with the aforementioned clear future work, there are still some open questions related to the personalized IV applications worthy discussing:

- *Balance between personalization and generalization*: Is more personalization always better? What level of personalization is sufficient, and how do we decide this? Is there a risk of indulging drivers too much with personalized features? Where should we set the limit?
- *Robustness of Personalized Models*: How robust are personalized products? Given that personalization might lead to overfitting, how well will our models perform in rare or unseen scenarios?

By addressing these points of discussion, we aim to delve deeper into the complexity and challenges of personalized behavior modeling in the context of intelligent vehicles.

BIBLIOGRAPHY

[1] H. & Company, "How many cars are there in the world in 2023?" 2023. [Online]. Available: https://hedgescompany.com/blog/2021/06/how-many-cars-are-there-in-the-world/

[2] T. Stewart, "Overview of Motor Vehicle Traffic Crashes in 2021," National Center for Statistics and Analysis, National Highway Traffic Safety Administration, DOT HS 813 435, Apr. 2023.

[3] INRIX, "INRIX: Congestion Costs Each American Nearly 100 hours, $1,400 A Year." 2020. [Online]. Available: https://inrix.com/press-releases/2019-traffic-scorecard-us

[4] U.S. Department of Energy, "Fuel wasted in traffic congestion." 2015.

[5] Nevada Department of Transportation, "Automated and Connected Vehicles." 2023. [Online]. Available: https://www.dot.nv.gov/mobility/avcv

[6] SAE International On-Road Automated Driving (ORAD) committee, *Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems*. 2014. doi: https://doi.org/10.4271/J3016_201401.

[7] J. B. Kenney, "Dedicated Short-Range Communications (DSRC) Standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011, doi: 10.1109/JPROC.2011.2132790.

[8] C. Bettisworth *et al.*, "Status of the Dedicated Short-Range Communications Technology and Applications: Report to Congress," John A. Volpe National Transportation Systems Center (U.S.), Technical Report FHWA-JPO-15-218, Jul. 2015. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/3575

[9] Z. Wang, Y. Bian, S. E. Shladover, G. Wu, S. E. Li, and M. J. Barth, "A survey on cooperative longitudinal motion control of multiple connected and automated vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 1, pp. 4–24, 2020.

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

[11] Z. Wang *et al.*, "Driver Behavior Modeling Using Game Engine and Real Vehicle: A Learning-Based Approach," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 738–749, Dec. 2020, doi: 10.1109/TIV.2020.2991948.

[12] D. Li and A. Liu, “Personalized highway pilot assist considering leading vehicle’s lateral behaviors,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, p. 095440702210811, Feb. 2022, doi: 10.1177/09544070221081190.

[13] L. Sun, W. Zhan, and M. Tomizuka, “Probabilistic Prediction of Interactive Driving Behavior via Hierarchical Inverse Reinforcement Learning,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC).*, 2018, pp. 2111–2117. doi: 10.0/Linux-x86_64.

[14] A. Wahab, C. Quek, C. K. Tan, and K. Takeda, “Driving profile modeling and recognition based on soft computing approach,” *IEEE transactions on neural networks*, vol. 20, no. 4, pp. 563–582, 2009.

[15] A. E. Kazdin, A. P. Association, and others, *Encyclopedia of psychology*, vol. 2. American Psychological Association Washington, DC, 2000.

[16] G. R. VandenBos, *APA dictionary of psychology*. American Psychological Association, 2007.

[17] C. M. Martinez, M. Heucke, F.-Y. Wang, B. Gao, and D. Cao, “Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 666–676, 2017.

[18] R. P. Payyanadan and L. S. Angell, “A Framework for Building Comprehensive Driver Profiles,” *Information*, vol. 13, no. 2, p. 61, 2022.

[19] X. Wu, Y. Gorospe, T. Misu, Y. Huynh, and N. Guerrero, “What Driving Says About You: A Small-Sample Exploratory Study Between Personality and Self-Reported Driving Style Among Young Male Drivers,” in *12th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 2020, pp. 104–110.

[20] T. Zimasa, S. Jamson, and B. Henson, “The influence of driver’s mood on car following and glance behaviour: Using cognitive load as an intervention,” *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 66, pp. 87–100, Oct. 2019, doi: 10.1016/j.trf.2019.08.019.

[21] R. D. Garrity and J. Demick, “Relations among personality traits, mood states, and driving behaviors,” *Journal of Adult Development*, vol. 8, no. 2, pp. 109–118, 2001.

[22] C. Rolim and P. Baptista, “Comparing drivers’ self-perception on driving behaviour changes with real world driving performance data: Lisbon case-study,” *Travel behaviour and society*, vol. 11, pp. 86–92, 2018.

[23] K.-T. Chen and H.-Y. W. Chen, “Driving style clustering using naturalistic driving data,” *Transportation research record*, vol. 2673, no. 6, pp. 176–188, 2019.

[24] T. Bär, D. Nienhüser, R. Kohlhaas, and J. M. Zöllner, “Probabilistic driving style determination by means of a situation based analysis of the vehicle data,” in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2011, pp. 1698–1703.

[25] Y. Lei, K. Liu, Y. Fu, X. Li, Z. Liu, and S. Sun, “Research on driving style recognition method based on driver’s dynamic demand,” *Advances in Mechanical Engineering*, vol. 8, no. 9, pp. 1–14, Sep. 2016, doi: 10.1177/1687814016670577.

[26] V. Vaitkus, P. Lengvenis, and G. Žylius, “Driving style classification using long-term accelerometer information,” in *2014 19th International Conference on Methods and Models in Automation and Robotics, MMAR 2014*, Institute of Electrical and Electronics Engineers Inc., Nov. 2014, pp. 641–644. doi: 10.1109/MMAR.2014.6957429.

[27] R. Stoichkov, “Android smartphone application for driving style recognition,” *Department of Electrical Engineering and Information Technology Institute for Media Technology*, 2013.

[28] K. S. Refaat, K. Ding, N. Ponomareva, and S. Ross, “Agent Prioritization for Autonomous Navigation.” arXiv, Sep. 18, 2019. Accessed: Jan. 20, 2023. [Online]. Available: http://arxiv.org/abs/1909.08792

[29] A. Vaswani *et al.*, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[30] E. Leurent and J. Mercat, “Social Attention for Autonomous Decision-Making in Dense Traffic.” arXiv, Nov. 27, 2019. Accessed: Dec. 19, 2022. [Online]. Available: http://arxiv.org/abs/1911.12250

[31] D. Cao, J. Li, H. Ma, and M. Tomizuka, “Spectral temporal graph neural network for trajectory prediction,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 1839–1845.

[32] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, “Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14424–14432.

[33] O. Makansi *et al.*, “You mostly walk alone: Analyzing feature attribution in trajectory prediction,” *arXiv preprint arXiv:2110.05304*, 2021.

[34] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, H. Rezatofighi, and S. Savarese, "Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[35] W. Wang, L. Wang, C. Zhang, C. Liu, and L. Sun, "Social Interactions for Autonomous Driving: A Review and Perspectives," 2022, doi: 10.48550/ARXIV.2208.07541.

[36] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 50, pp. 24972–24978, 2019.

[37] N. Buckman, A. Pierson, W. Schwarting, S. Karaman, and D. Rus, "Sharing is caring: Socially-compliant autonomous intersection negotiation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 6136–6143.

[38] Z. Zhao *et al.*, "Personalized Car Following for Autonomous Driving with Inverse Reinforcement Learning," in *Proceedings 2022 IEEE International Conference on Robotics and Automation.*, 2022. [Online]. Available: https://www.researchgate.net/publication/358977124

[39] X. Liao *et al.*, "Online Prediction of Lane Change with a Hierarchical Learning-Based Approach," in *Proceedings 2022 IEEE International Conference on Robotics and Automation*, Philadelphia, 2022. [Online]. Available: https://www.researchgate.net/publication/358885848

[40] M. Hasenjager and H. Wersing, "Personalization in advanced driver assistance systems and autonomous vehicles: A review," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama: IEEE, Oct. 2017, pp. 1–7. doi: 10.1109/ITSC.2017.8317803.

[41] S. Lefèvre, A. Carvalho, Y. Gao, H. E. Tseng, and F. Borrelli, "Driver models for personalised driving assistance," *Vehicle System Dynamics*, vol. 53, no. 12, pp. 1705–1720, Dec. 2015, doi: 10.1080/00423114.2015.1062899.

[42] I. Bae *et al.*, "Self-driving like a human driver instead of a robocar: Personalized comfortable driving experience for autonomous vehicles," *arXiv preprint arXiv:2001.03908*, 2020.

[43] W. Wang, D. Zhao, W. Han, and J. Xi, "A Learning-Based Approach for Lane Departure Warning Systems With a Personalized Driver Model," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9145–9157, Oct. 2018, doi: 10.1109/TVT.2018.2854406.

[44] W. Wang, J. Xi, and J. K. Hedrick, “A Learning-Based Personalized Driver Model Using Bounded Generalized Gaussian Mixture Models,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11679–11690, Dec. 2019, doi: 10.1109/TVT.2019.2948911.

[45] Z. Huang, J. Wu, and C. Lv, “Driving Behavior Modeling Using Naturalistic Human Driving Data With Inverse Reinforcement Learning,” *IEEE Transactions on Intelligent Transportation Systems*, 2021, doi: 10.1109/TITS.2021.3088935.

[46] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, and others, “Maximum entropy inverse reinforcement learning.,” in *Aaai*, Chicago, IL, USA, 2008, pp. 1433–1438.

[47] M. Naumann, L. Sun, W. Zhan, and M. Tomizuka, “Analyzing the Suitability of Cost Functions for Explaining and Imitating Human Driving Behavior based on Inverse Reinforcement Learning,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 5481–5487.

[48] C. Katrakazas, M. Quddus, W. H. Chen, and L. Deka, “Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions,” *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015.

[49] A. Carvalho, S. Lefevre, G. Schildbach, J. Kong, and F. Borrelli, “Automated driving: The role of forecasts and uncertainty: A control perspective,” *European Journal of Control*, vol. 24, pp. 14–32, 2015.

[50] K. Berntorp and S. Di Cairano, “Joint decision making and motion planning for road vehicles using particle filtering,” *IFAC-PapersOnLine*, vol. 49, no. 11, pp. 175–181, 2016.

[51] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, “A review of motion planning for highway autonomous driving,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2019.

[52] Y. Xing *et al.*, “Driver lane change intention inference for intelligent vehicles: framework, survey, and challenges,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4377–4390, 2019.

[53] K. Schmidt, M. Beggiato, K. H. Hoffmann, and J. F. Krems, “A mathematical model for predicting lane changes using the steering wheel angle,” *Journal of safety research*, vol. 49, pp. 85–90, 2014.

[54] L. Y. Du, W. Chen, J. Ji, Z. H. Pei, B. M. Tong, and H. J. Zheng, “A novel intelligent approach to lane-change behavior prediction for intelligent and connected vehicles,” *Computational Intelligence and Neuroscience*, vol. 2022, p. Article ID 9516218, 2022.

[55] F. Li, W. Wang, G. Feng, and W. Guo, “Driving intention inference based on dynamic bayesian networks,” in *Practical applications of intelligent systems*, 2014, pp. 1109–1119.

[56] Z. Shou, Z. Wang, K. Han, Y. Liu, P. Tiwari, and X. Di, “Long-term prediction of lane change maneuver through a multilayer perceptron,” in *IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2020.

[57] J. Ashesh, H. Koppula, S. Soh, B. Raghavan, A. Singh, and A. Saxena, “Brain4cars: Car that knows before you do via sensory-fusion deep learning architecture,” *arXiv preprint arXiv:1601.00740*, 2016.

[58] O. Scheel, N. S. Nagaraja, L. Schwarz, N. Navab, and F. Tombari, “Attention-based lane change prediction,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8655–8661.

[59] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, “Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1672–1678.

[60] Q. Xue, Y. Xing, and J. Lu, “An integrated lane change prediction model incorporating traffic context based on trajectory data,” *Transportation Research Part C: Emerging Technologies*, vol. 141, p. 103738, 2022.

[61] T. Awal, L. Kulik, and K. Ramamohanrao, “Optimal traffic merging strategy for communication- and sensor-enabled vehicles,” in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, 2013, pp. 1468–1474.

[62] R. Scarinci, A. Hegyi, and B. Heydecker, “Definition of a merging assistant strategy using intelligent vehicles,” *Transportation Research Part C: Emerging Technologies*, vol. 82, pp. 161–179, Sep. 2017.

[63] X. Y. Lu and J. K. Hedrick, “Longitudinal control algorithm for automated vehicle merging,” *International Journal of Control*, vol. 76, no. 2, pp. 193–202, Jan. 2003.

[64] V. Jain, D. Liu, and S. Baldi, “Adaptive strategies to platoon merging with vehicle engine uncertainty,” in *IFAC-PapersOnline*, 2020, pp. 15065–15070.

[65] S. Jing, F. Hui, X. Zhao, J. Rios-Torres, and A. J. Khattak, “Cooperative Game Approach to Optimal Merging Sequence and on-Ramp Merging Control of Connected and Automated Vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4234–4244, Nov. 2019, doi: 10.1109/TITS.2019.2925871.

[66] S. Akti, I. G. Erdagi, M. A. Silgu, and H. B. Celikoglu, “A Game-Theoretical Approach for Lane-Changing Maneuvers on Freeway Merging Segments,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020*, Institute of Electrical and Electronics Engineers Inc., Sep. 2020. doi: 10.1109/ITSC45102.2020.9294458.

[67] M. Wang, S. P. Hoogendoorn, W. Daamen, B. van Arem, and R. Happee, “Game theoretic approach for predictive lane-changing and car-following control,” *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 73–92, Sep. 2015.

[68] Q. Zhang, D. Filev, H. E. Tseng, S. Szwabowski, and R. Langari, “Addressing Mandatory Lane Change Problem with Game Theoretic Model Predictive Control and Fuzzy Markov Chain,” in *Proceedings of the American Control Conference*, Institute of Electrical and Electronics Engineers Inc., Aug. 2018, pp. 4764–4771. doi: 10.23919/ACC.2018.8431530.

[69] H. Min and others, “On-ramp merging strategy for connected and automated vehicles based on complete information static game,” *Journal of Traffic and Transportation Engineering*, vol. 8, no. 4, pp. 582–595, 2021.

[70] M. Fellendorf and P. Vortisch, *Microscopic Traffic Flow Simulator VISSIM*, vol. 145. in International Series in Operations Research & Management Science, vol. 145. 2010.

[71] Aimsun, *Aimsun Next 20 User’s Manual, Aimsun Next Version 20.0.3*. Barcelona, Spain, 2021.

[72] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “SUMO simulation of urban mobility an overview,” in *Proceedings of the 3rd international conference on advances in system simulation*, Oct. 2011.

[73] P. A. Lopez *et al.*, “Microscopic Traffic Simulation using SUMO,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, 2018, pp. 2575–2582.

[74] G. Rong *et al.*, “LGSVL Simulator: A High Fidelity Simulator for Autonomous Driving,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[75] T. Case, “The Use of Emerging Virtual Reality Technology in Road Safety Analysis: The Hook-Turn Case,” in *99th Annual Meeting of the Transportation Research Board*, Washington, D.C., Jan. 2020.

[76] D. Jia, J. Sun, A. Sharma, Z. Zheng, and B. Liu, “Integrated simulation platform for conventional, connected and automated driving: A design from cyber–physical

systems perspective," *Transportation Research Part C: Emerging Technologies*, vol. 124, p. 102984, Mar. 2021.

[77] C. Biurrun, L. Serrano-Arriezu, and C. Olaverri-Monreal, "Microscopic Driver-Centric Simulator: Linking Unity3D and SUMO," in *Recent Advances in Information Systems and Technologies*, in Advances in Intelligent Systems and Computing. Cham, Switzerland: Springer International Publishing AG, Mar. 2017, pp. 851–860.

[78] G. Schirner, D. Erdogmus, K. Chowdhury, and T. Padir, "The Future of Human-in-the-Loop Cyber-Physical Systems," *Computer*, vol. 46, no. 1, Jan. 2013.

[79] J. B.-C. Chen and H. Peng, "Differential-Braking-Based Rollover Prevention for Sport Utility Vehicles with Human-in-the-loop Evaluations," *Vehicle System Dynamics*, vol. 36, no. 4, pp. 359–389, 2001, doi: 10.1076/vesd.36.4.359.3546.

[80] W. Li, D. Sadigh, S. S. Sastry, and S. A. Seshia, "Synthesis for Human-in-the-Loop Control Systems," in *Tools and Algorithms for the Construction and Analysis of Systems*, Springer, 2014. doi: 10.1007/978-3-642-54862-8_40.

[81] M. Szalai, B. Varga, T. Tettamanti, and V. Tihanyi, "Mixed reality test environment for autonomous cars using Unity 3D and SUMO," in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2020, pp. 73–78. doi: 10.1109/SAMI48414.2020.9108745.

[82] Grand View Research, "Digital Twin market size, share & trends analysis report by end-use (automotive & transport, retail & consumer goods, agriculture, manufacturing, energy & utilities), by region, and segment forecasts, 2021 - 2028." 2021. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/digital-twin-market

[83] Z. Wang *et al.*, "Mobility Digital Twin: Concept, Architecture, Case Study, and Future Challenges," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 17452–17467, Mar. 2022.

[84] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and US Air Force vehicles," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, 2012, p. 1818.

[85] X. Chen, E. Kang, S. Shiraishi, V. M. Preciado, and Z. Jiang, "Digital behavioral twins for safe connected cars," in *Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, 2018, pp. 144–153.

[86] Z. Wang *et al.*, "A Digital Twin paradigm: Vehicle-to-Cloud based advanced driver assistance systems," in *2020 IEEE 91st Vehicular Technology Conference*, 2020, pp. 1–6.

[87] X. Liao *et al.*, "Cooperative Ramp Merging Design and Field Implementation: A Digital Twin Approach Based on Vehicle-to-Cloud Communication," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4490–4500, May 2022, doi: 10.1109/TITS.2020.3045123.

[88] Y. Liu, Z. Wang, K. Han, Z. Shou, P. Tiwari, and J. Hansen, "Vision-cloud data fusion for ADAS: A lane change prediction case study," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, Jun. 2022.

[89] Z. Wang, K. Han, and P. Tiwari, "Digital Twin-assisted cooperative driving at non-signalized intersections," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, Jun. 2022.

[90] Z. Wang, K. Han, and P. Tiwari, "Digital twin simulation of connected and automated vehicles with the unity game engine," in *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*, 2021, pp. 1–4.

[91] C. Schwarz and Z. Wang, "The role of digital twins in connected and automated vehicles," *IEEE Intelligent Transportation Systems Magazine*, pp. 2–11, Jan. 2022.

[92] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2009, pp. 336–343.

[93] X. Zhao *et al.*, "Co-Simulation Platform for Modeling and Evaluating Connected and Automated Vehicles and Human Behavior in Mixed Traffic," *SAE MobilityRxiv™*, Nov. 2021.

[94] J. Song, Y. Wu, Z. Xu, and X. Lin, "Research on car-following model based on SUMO," in *The 7th IEEE/International Conference on Advanced Infocomm Technology*, Fuzhou, 2014, pp. 47–55.

[95] J. Erdmann, "Lane-Changing Model in SUMO," *Proceedings of the SUMO2014 modeling mobility with open data*, vol. 24, pp. 77–88, 2014.

[96] F. de Ponte Muller, "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles," *Sensors*, vol. 17, no. 2, p. 271, 2017.

[97] E. Siedlecka and T. F. Denson, "Experimental methods for inducing basic emotions: A qualitative review," *Emotion Review*, vol. 11, no. 1, pp. 87–97, 2019.

[98] R. R. McCrae, P. T. Costa Jr, and T. A. Martin, "The NEO–PI–3: A more readable revised NEO personality inventory," *Journal of personality assessment*, vol. 84, no. 3, pp. 261–270, 2005.

[99] R. R. McCrae and O. P. John, "An introduction to the five-factor model and its applications," *Journal of personality*, vol. 60, no. 2, pp. 175–215, 1992.

[100] P. Costa and R. McCrae, "NEO inventories professional manual: NEO-PI-3," NEO-FFI-3, NEO PI-R, 2010.

[101] P. C. Terry, A. M. Lane, and G. J. Fogarty, "Construct validity of the Profile of Mood States—Adolescents for use with adults," *Psychology of sport and exercise*, vol. 4, no. 2, pp. 125–139, 2003.

[102] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*, Springer, 2018, pp. 621–635.

[103] Q. Xue, K. Wang, J. J. Lu, and Y. Liu, "Rapid driving style recognition in car-following using machine learning and vehicle trajectory data," *Journal of advanced transportation*, vol. 2019, 2019.

[104] J. Chen, D. Sun, M. Zhao, and Y. Li, "DCFS-based online driving preferences learning approach with application to personalized lane keeping controller design," *International journal of automotive technology*, vol. 22, pp. 1373–1385, 2021.

[105] M. Almallah, R. Alfahel, Q. Hussain, W. K. Alhajyaseen, and C. Dias, "Empirical evaluation of drivers' start-up behavior at signalized intersection using driving simulator," *Procedia Computer Science*, vol. 170, pp. 227–234, 2020.

[106] E. R. Khansari, M. Tabibi, and F. M. Nejad, "A study on following behavior based on the time headway," *Jurnal Kejuruteraan*, vol. 32, no. 2, pp. 187–195, 2020.

[107] V. L. Knoop, S. P. Hoogendoorn, Y. Shiomi, and C. Buisson, "Quantifying the Number of Lane Changes in Traffic An empirical analysis," in *TRB 2016, Transportation Research Board 95th annual meeting*, 2016, pp. 16-p.

[108] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[109] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[110] M. Pal, "Random forest classifier for remote sensing classification," *International journal of remote sensing*, vol. 26, no. 1, pp. 217–222, 2005.

[111] M. Grandini, E. Bagli, and G. Visani, “Metrics for multi-class classification: an overview,” *arXiv preprint arXiv:2008.05756*, 2020.

[112] T. A. Widiger and J. R. Oltmanns, “Neuroticism is a fundamental domain of personality with enormous public health implications,” *World psychiatry*, vol. 16, no. 2, p. 144, 2017.

[113] Y. Wang, W. Qu, Y. Ge, X. Sun, and K. Zhang, “Effect of personality traits on driving style: Psychometric adaption of the multidimensional driving style inventory in a Chinese sample,” *PLoS one*, vol. 13, no. 9, p. e0202126, 2018.

[114] M. Liang *et al.*, “Learning Lane Graph Representations for Motion Forecasting.” arXiv, Jul. 27, 2020. Accessed: Dec. 20, 2022. [Online]. Available: http://arxiv.org/abs/2007.13732

[115] Y. Yuan, X. Weng, Y. Ou, and K. Kitani, “AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 9793–9803. doi: 10.1109/ICCV48922.2021.00967.

[116] J. Ngiam *et al.*, “Scene Transformer: A unified architecture for predicting multiple agent trajectories.” arXiv, Mar. 04, 2022. Accessed: Dec. 19, 2022. [Online]. Available: http://arxiv.org/abs/2106.08417

[117] G. Markkula *et al.*, “Defining interactions: a conceptual framework for understanding interactive behaviour in human and automated road traffic,” *Theoretical Issues in Ergonomics Science*, vol. 21, no. 6, pp. 728–752, Nov. 2020, doi: 10.1080/1463922X.2020.1736686.

[118] C. W. Granger, “Investigating causal relations by econometric models and cross-spectral methods,” *Econometrica: journal of the Econometric Society*, pp. 424–438, 1969.

[119] J. M. McCracken, “Exploratory causal analysis with time series data,” *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 1–147, 2016.

[120] A. Shojaie and E. B. Fox, “Granger Causality: A Review and Recent Advances,” *Annu. Rev. Stat. Appl.*, vol. 9, no. 1, pp. 289–319, Mar. 2022, doi: 10.1146/annurev-statistics-040120-010930.

[121] R. Nathan *et al.*, “A movement ecology paradigm for unifying organismal movement research,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 49, pp. 19052–19059, 2008.

[122] K. Fujii *et al.*, “Learning interaction rules from multi-animal trajectories via augmented behavioral models,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 11108–11122, 2021.

[123] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, “Optimal trajectory generation for dynamic street scenarios in a frenet frame,” in *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 987–993.

[124] R. Marcinkevicˇs and J. E. Vogt, “INTERPRETABLE MODELS FOR GRANGER CAUSALITY USING SELF-EXPLAINING NEURAL NETWORKS,” p. 23, 2021.

[125] A. Tank, I. Covert, N. Foti, A. Shojaie, and E. Fox, “Neural granger causality for nonlinear time series,” *stat*, vol. 1050, p. 16, 2018.

[126] A. B. Barrett and L. Barnett, “Granger causality is designed to measure effect, not mechanism,” *Front. Neuroinform.*, vol. 7, 2013, doi: 10.3389/fninf.2013.00006.

[127] D. Alvarez Melis and T. Jaakkola, “Towards robust interpretability with self-explaining neural networks,” *Advances in neural information processing systems*, vol. 31, 2018.

[128] L. Li, J. Gan, K. Zhou, X. Qu, and B. Ran, “A novel lane-changing model of connected and automated vehicles: Using the safety potential field theory,” *Physica A: Statistical Mechanics and its Applications*, vol. 559, p. 125039, Dec. 2020, doi: 10.1016/j.physa.2020.125039.

[129] C. D. of M. Vehicles, “Vehicle Positioning.” 2021. [Online]. Available: https://www.dmv.ca.gov/portal/es/handbook/california-driver-handbook/vehicle-positioning/

[130] W. B. Nicholson, D. S. Matteson, and J. Bien, “VARX-L: Structured regularization for large vector autoregressions with exogenous variables,” *International Journal of Forecasting*, vol. 33, no. 3, pp. 627–651, 2017.

[131] W. Zhan *et al.*, “Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps,” *arXiv preprint arXiv:1910.03088*, 2019.

[132] X. Liao *et al.*, “Driver Digital Twin for Online Prediction of Personalized Lane Change Behavior,” *IEEE Internet of Things Journal*, 2023.

[133] X. Liao, S. Mehrotra, S. Ho, Y. Gorospe, X. Wu, and T. Misu, “Driver Profile Modeling Based on Driving Style, Personality Traits, and Mood States,” in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022.

[134] X. Liao, Z. Wang, K. Han, X. Zhao, M. Barth, and G. Wu, *Driver Behavior-Aware Cooperative Ramp Merging for Intelligent Vehicles*, vol. 3. Walter de Gruyter GmbH & Co KG, 2023.

[135] W. Xu, J. Wang, T. Fu, H. Gong, and A. Sobhani, "Aggressive driving behavior prediction considering drivers intention based on multivariate-temporal feature data," *Accident Analysis & Prevention*, vol. 164, p. 106477, 2022.

[136] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to Sequence Learning with Neural Networks," in *Neural Information Processing Systems*, 2014.

[137] S. Levine and V. Koltun, "Continuous Inverse Optimal Control with Locally Optimal Examples," 2012.

[138] S. Hochreiter and Jürgen Schmidhuber., "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[139] F. Altche and A. de L. Fortelle, "An LSTM network for highway trajectory prediction," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 353–359.

[140] X. Luo, D. Li, Y. Yang, and S. Zhang, "Spatiotemporal traffic flow prediction with KNN and LSTM," *Journal of Advanced Transportation*, vol. 2019, 2019.

[141] W. Kim, Y. Han, K. J. Kim, and K.-W. Song, "Electricity load forecasting using advanced feature selection and optimal deep learning model for the variable refrigerant flow systems," *Energy Reports*, vol. 6, pp. 2604–2618, 2020.

[142] V. Mahajan, C. Katrakazas, and C. Antoniou, "Prediction of lane-changing maneuvers with automatic labeling and deep learning," *Transportation research record*, vol. 2674, no. 7, pp. 336–347, 2020.

[143] R. Van Den Boomgaard and R. Van Balen, "Methods for fast morphological image transforms using bitmapped binary images," *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 3, pp. 252–258, 1992.

[144] J. Quehl, H. Hu, O. S. Tas, E. Rehder, and M. Lauer, "How good is my prediction? Finding a similarity measure for trajectory prediction evaluation," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6.

[145] X. Liao *et al.*, "Game Theory-Based Ramp Merging for Mixed Traffic With Unity-SUMO Co-Simulation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021, doi: 10.1109/TSMC.2021.3131431.

[146] N. Isa, M. Yusoff, and A. Mohamed, “A Review on Recent Traffic Congestion Relief Approaches,” in *Proceedings - 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology, ICAIET 2014*, Institute of Electrical and Electronics Engineers Inc., Dec. 2015, pp. 121–126. doi: 10.1109/ICAIET.2014.29.

[147] J. Rios-Torres and A. A. Malikopoulos, “A Survey on the Coordination of Connected and Automated Vehicles at Intersections and Merging at Highway On-Ramps,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, May 2017, doi: 10.1109/TITS.2016.2600504.

[148] R. Scarinci and B. Heydecker, “Control Concepts for Facilitating Motorway On-ramp Merging Using Intelligent Vehicles,” *Transport Reviews*, vol. 34, no. 6, pp. 775–797, Nov. 2014, doi: 10.1080/01441647.2014.983210.

[149] Z. Zhao, Z. Wang, G. Wu, F. Ye, and M. Barth, “The State-of-the-Art of Coordinated Ramp Control with Mixed Traffic Conditions,” in *2019 IEEE 22nd International Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2019, pp. 1–6.

[150] H. Yu, H. E. Tseng, and R. Langari, “A human-like game theory-based controller for automatic lane changing,” *Transportation Research Part C: Emerging Technologies*, vol. 88, pp. 140–158, Mar. 2018, doi: 10.1016/j.trc.2018.01.016.

[151] C. Dong, J. M. Dolan, and B. Litkouhi, “Intention Estimation For Ramp Merging Control In Autonomous Driving,” in *2017 IEEE intelligent vehicles symposium (IV)*, 2017, pp. 1584–1589.

[152] I. Mayergoyz, *Mathematical Models of Hysteresis and Their Applications*. New York, NY, USA: Elsevier, 2003.

[153] Z. Wang, G. Wu, and M. J. Barth, “Developing a distributed consensus-based cooperative adaptive cruise control (CACC) system,” in *Transportation Research Board 96th Annual Meeting*, Jan. 2017.

[154] K. Kang and H. A. Rakha, “A Repeated Game Freeway Lane Changing Model,” *Sensors*, vol. 20, no. 6, p. 1554, Mar. 2020.

[155] E. Brockfeld, R. D. Kuhne, and P. Wagner, “Calibration and validation of microscopic traffic flow models,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1876, pp. 62–70, 2004.

[156] M. Zhu, X. Wang, A. Tarko, and S. Fang, “Modeling car-following behavior on urban expressways in shanghai: A naturalistic driving study,” *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 425–445, Aug. 2018.

[157] E. Bj"arkvik, F. F"urer, M. Pourabdollah, and B. Lindenberg, “Simulation and characterisation of traffic on drive me route around gothenburg using SUMO,” in *Proc. Towards Simulation Auton. Mobility (SUMO)*, 2017, pp. 1–3.

[158] Z. Wang, G. Wu, and G. Scora, “MOVESTAR: An open-source vehicle fuel and emission model based on USEPA MOVES,” arXiv preprint arXiv:2008.04986v2, 2020.

[159] M. Kamrani, R. Arvin, and A. J. Khattak, “Extracting useful information from basic safety message data: An empirical study of driving volatility measures and crash frequency at intersections,” *Transportation Research Record*, vol. 2672, no. 38, pp. 290–301, Dec. 201AD.

[160] J. Piao and M. McDonald, “Advanced driver assistance systems from autonomous to cooperative approach,” *Transport reviews*, vol. 28, no. 5, pp. 659–684, 2008.

[161] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, “Survey of pedestrian detection for advanced driver assistance systems,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, 2009.

[162] G. Hegeman, K. Brookhuis, and S. Hoogendoorn, “Opportunities of advanced driver assistance systems towards overtaking,” *European journal of transport and infrastructure research EJTIR, 5 (4)*, 2005.

[163] A. Vahidi and A. Eskandarian, “Research advances in intelligent collision avoidance and adaptive cruise control,” *IEEE transactions on intelligent transportation systems*, vol. 4, no. 3, pp. 143–153, 2003.

[164] N. Van Nes, M. Houtenbos, and I. Van Schagen, “Improving speed behaviour: the potential of in-car speed assistance and speed limit credibility,” *IET Intelligent Transport Systems*, vol. 2, no. 4, pp. 323–330, 2008.

[165] S. Vlassenroot, E. Molin, D. Kavadias, V. Marchau, K. Brookhuis, and F. Witlox, “What drives the acceptability of intelligent speed assistance (ISA)?,” *European journal of transport and infrastructure research.-Delft*, vol. 11, no. 2, pp. 256–273, 2011.

[166] X. Qi, P. Wang, G. Wu, K. Boriboonsomsin, and M. J. Barth, “Connected cooperative ecodriving system considering human driver error,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2721–2733, Aug. 2018, doi: 10.1109/TITS.2018.2845799.

[167] Z. Constantinescu, C. Marinoiu, and M. Vladoiu, “Driving style analysis using data mining techniques,” *International Journal of Computers Communications & Control*, vol. 5, no. 5, pp. 654–663, 2010.

[168] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[169] A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks: Prediction and system modelling," LA-UR-87-2662; CONF-8706130-4; ON: DE88006479, 1987.

[170] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[171] U.S. Department of Transportation, "Wyoming DOT (WYDOT) Connected Vehicle Pilot Determines Appropriate Tractor-Trailer Antenna Placement and Equipment Configuration." 2020. [Online]. Available: https://www.its.dot.gov/pilots/wyoming_antenna.htm