Simulating Naturalistic Driving Environment for Autonomous Vehicles

by

Xintao Yan

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering and Scientific Computing)
in the University of Michigan
2023

Doctoral Committee:

Professor Henry X. Liu, Chair
Associate Professor Neda Masoud
Associate Professor Ramanarayan Vasudevan
Professor Yafeng Yin

Xintao Yan

xintaoy@umich.edu

ORCID iD: 0000-0002-0569-5628

To my entire family for their boundless and unconditional love.

ACKNOWLEDGMENTS

I extend my sincere gratitude to my advisor, Professor Henry X. Liu, for his consistent guidance, encouragement, and support throughout every stage of my Ph.D. journey. I can still vividly recall the excitement I felt in Dr. Liu's Didi office in 2017 when he offered me the opportunity to join his research group. Over the years, this incredible voyage has exceeded my wildest expectations and, although challenging, has provided me with unparalleled enrichment and rewards. Dr. Liu not only taught me how to do research but more importantly, the importance of identifying critical real-world problems from the first principle. He consistently set the bar high, teaching us to produce impactful work and cultivating my academic discernment. His approach to academic rigor, genuine curiosity about hardcore research questions, and unwavering dedication have left an indelible mark on my character, shaping my path for future endeavors. Beyond the academic sphere, Dr. Liu has imparted the importance of family, providing support and care for my personal life which I deeply appreciate. I think no amount of words can express my feelings, his wisdom, professionalism, and kindness in work and life have set a lifetime model for me. I am truly honored to be his student.

I would also like to thank my committee members, Professors Yafeng Yin, Neda Masoud, and Ramanarayan Vasudevan, for their valuable and constructive suggestions throughout my doctoral study. I appreciate the opportunities to learn and work with them in classes, seminars, and projects. Particular thanks go to Dr. Yin and Dr. Masoud for creating such a friendly and supportive environment for the NGTS program and the entire department.

Special thanks to my collaborators and friends, Professors Shuo Feng, Yiheng Feng, and Zhengxia Zou. I would like to thank Dr. Shuo Feng for his help and support during my Ph.D. study. I enjoyed all the discussions we had on various projects. I am very grateful to Dr. Yiheng Feng for his guidance during my summer internship, which makes me determined to pursue my Ph.D. at UMich. I would like to thank Dr. Zhengxia Zou and I learned a lot during the discussions and collaborations with him.

I would like to thank my former and current lab mates in Michigan Traffic Lab, Prof. Shuo Feng, Prof. Yiheng Feng, Prof. Wai Wong, Prof. Zhengxia Zou, Prof. Xiaowei Shi, Dr. Weili Sun, Dr. Jianfeng Zheng, Dr. Yan Zhao, Dr. Shihong Huang, Dr. Zhen Yang, Dr. Lin Liu, Dr. Rusheng Zhang, Dr. Depu Meng, Dr. Ronan Keane, Dr. Yukun Zuo, Dr. Ran Sun, Dr. Jiawei Wang, Dr. Yuanxin Zhong, Dr. Boqi Li, Dr. Tinghan Wang, Dr. Xingmin Wang, Shengyin Shen, Haowei Sun, Haojie Zhu, Zihao Wang, Zachary Jerome, and Zhijie Qiao. Especially, I want to thank Xingmin, Shengyin, and Yuanxin for their support over the past five years.

I want to extend my thanks to all my friends, including but not limited to Prof. Zhengtian Xu, Prof. Xiaotong Sun, Prof. Ziyou Song, Prof. Long Zhang, Prof. Zhaojian Li, Prof. Yulin Pan, Dr. Yutong Li, Dr. Qiuhao Hu, Dr. Niankai Yang, Dr. Kaiwen Liu, Xiang Wei, Weifan Zhang, Zheyu Zhang, Zongyang Hu, Xiao Li, Jun Zhou, Yaozhong Liu, Zhou Zhang, Lu Wen, Zaiyi Jiang, Sitong Liu, Wenshan Yu, Ling Lu, Zhihuang Zhang, Changyao Huang, Siyuan Feng, Xieyuan Zhang. It is all of you making my life in Ann Arbor so colorful and enjoyable. I will never forget those moments during our travels, basketball games, and dinner parties. In particular, I'd like to express

my gratitude to Xieyuan Zhang and Xiang Wei for our cherished friendship. Xieyuan, from Beijing to Ann Arbor, it is always a pleasure to see you.

Finally, I must convey my heartfelt appreciation to my fiancée, Jing Rong, for her companionship and love throughout the past years. The best thing in the world is having you by my side. Simultaneously, I owe my deepest gratitude to my family, particularly my parents, Zhihui Yan and Caixia He, for their unwavering support and boundless love. I owe my achievements to them entirely.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

LIST OF ACRONYMS

AI Artificial Intelligence

AV Autonomous Vehicle

BV Background Vehicle

DAS Data Acquisition System

DL Deep Learning

DRL Deep Reinforcement Learning

Delta-V Change in Velocity

FARS Fatality Analysis Reporting System

GAIL Generative Adversarial Imitation Learning

GAN Generative Adversarial Network

IDM Intelligent Driver Model

IL Imitation Learning

IS Importance Sampling

IVBSS Integrated Vehicle Based Safety System

KL-divergence Kullback–Leibler Divergence

LSTM Long Short-Term Memory

MC Monte Carlo

MDP Markov Decision Process

ML Machine Learning

MLP Multilayer Perceptron

MOBIL Minimizing Overall Braking Induced by Lane Change

NDE Naturalistic Driving Environment

NADE Naturalistic and Adversarial Driving Environment

NDD Naturalistic Driving Data

PET Post-Encroachment Time

POV Principal Other Vehicle

RHW Relative Half-Width

RL Reinforcement Learning

SM Surrogate Model

SPMD Safety Pilot Model Deployment

SUMO Simulation of Urban MObility

THW Time Headway

TTC Time-To-Collision

ABSTRACT

Autonomous Vehicle (AV) technology has the potential to revolutionize the future transportation landscape. Simulation plays an indispensable role in the development and testing of AVs, due to its unmatched advantages of controllability, repeatability, and cost-effectiveness. To build the simulation system, one key problem is how to model background agents behavior, in order to construct the Naturalistic Driving Environment (NDE). To ensure the trustworthiness of simulation results, the NDE must meet the novel and demanding criterion, i.e., statistical realism, in terms of simulation fidelity. The simulated NDE needs to be statistically representative of the real-world traffic environment, particularly for those long-tail safety-critical events, which are critical for AV safety. Unfortunately, the real-world NDE is spatiotemporally complex, highly interactive, and with only rare occurrences of safety-critical events. Therefore, how to build a high-fidelity simulator is a long-standing problem.

This dissertation aims to provide systematic methods for simulating high-fidelity NDE for AV development and testing, leveraging large-scale Naturalistic Driving Data (NDD). We first identify the statistical realism requirement on NDE, a new requirement for microscopic traffic simulators brought by AV applications. To achieve this, a data-driven method is proposed to characterize human car-following and lane-changing behavior distributions. The proposed method is validated using real-world data in the simulation of multi-lane highway driving environments.

In contrast to highway driving environments, urban environments usually involve more complex interactions between multiple agents. Therefore, we further develop NeualNDE, a Deep Learning (DL)-based NDE modeling framework. The results validate that NeuralNDE can achieve accurate normal driving statistics (e.g., vehicle speed/distance/yielding behavior distributions, etc.), and

more importantly, safety-critical driving statistics (e.g., crash rate/type/severity and near-miss statistics, etc.), as demonstrated in the simulation of real-world urban driving environments.

One important application of high-fidelity NDE is AV safety performance evaluation. We discover that sparse but adversarial adjustments to the NDE, resulting in the Naturalistic and Adversarial Driving Environment (NADE), can significantly reduce the required test miles without loss of evaluation unbiasedness. The results show that, compared with directly evaluating AV in NDE, the proposed NADE environment can accelerate the evaluation process by multiple orders of magnitude. Consequently, we provide a complete pipeline for accurate and efficient simulation-based AV testing.

In summary, this dissertation presents methodologies for building high-fidelity NDE and uses AV testing as an example to demonstrate the importance of NDE simulation. These proposed methods pave the way for enhancing AV safety performance, which is beneficial for all stakeholders, including AV developers, customers, and regulators, and contributes to the long-term development of AV technology.

CHAPTER 1

Introduction

1.1 Background

AV technology has emerged as a focal point for academia, industry, and government agencies due to its transformative potential in revolutionizing the future transportation landscape. The first is safety, in the US, 94% of fatal accidents are caused by human errors (National Highway Traffic Safety Administration, 2017). AVs hold the potential to significantly mitigate these issues, promising a remarkable enhancement in road safety (Jacobstein, 2019; Nunes and Axhausen, 2021). Beyond safety, AVs can serve as mobile actuators to dampen traffic oscillation and improve fuel efficiency (Wu et al., 2018; Stern et al., 2018). Furthermore, AVs possess the potential to alleviate traffic congestion (Guo et al., 2019; Yang et al., 2021), expand road capacity (Lu et al., 2020), enhance mobility accessibility (Andrews, 2023), and contribute positively to environmental sustainability (Taiebat et al., 2018; Kopelias et al., 2020). These compelling advantages position AV technology as a pivotal solution to the multifaceted challenges confronting our present transportation system.

In the last two decades, significant advancements have transpired in the field of AVs, primarily attributed to the swift evolution of computer technology and DL (LeCun et al., 2015). As early as 2015, several companies had declared their intentions to commence mass production of AVs by 2020. Regrettably, the actual progress has fallen short of these ambitious projections, and, as of now, no SAE Level 4 AVs (SAE International, 2021) are commercially available. Numerous factors contribute to this disparity between expectation and reality, with paramount among them being the fact that AVs' safety performance remains notably inferior to that of human drivers (Liu and Feng, 2022). For average drivers in the US, the occurrence probability of a crash is around 1.9×10^{-6} per mile (Kalra and Paddock, 2016). In contrast, the disengagement rate for the state-of-the-art AV is around 1.0×10^{-5} per mile, according to the 2022 Disengagement Reports from California (California Department of Motor Vehicles, 2023). Therefore, safety remains the key challenge for the AV technology (Feng et al., 2023).

Simulation provides a controllable, repeatable, efficient, and low-cost venue for both developing and testing AVs. For training purposes, simulation provides a controllable play field that can generate safety-critical events to train the full AV software stack. In this way, simulation can augment real-world training data and reduces reliance on them, which can be hard, costly, and hazardous to collect (Igl et al., 2022). For testing purposes, simulation offers an efficient and cost-effective alternative to real-world testing, significantly expediting the iterative development process and mitigating the substantial financial burdens associated with real-world experiments. Furthermore, simulation ensures scenario repeatability, enabling precise replication and consistent AV safety performance evaluation. Therefore, simulation is indispensable for AV development, and large AV companies rely on it extensively, for example, Waymo had simulated over 15 billion miles as of 2021 (Waymo LLC, 2021b).

1.2 Naturalistic driving environment simulation

1.2.1 Model human driving behavior

To build the simulation system, one key problem is how to model human driving behaviors, in order to construct NDE. The simulated NDE needs to reproduce the real-world traffic environment, with diverse, nuanced, and long-tailed microscopic human driving behaviors. In recent years, great efforts have been made in developing simulators for AV. Thanks to rapid advances in Artificial Intelligence (AI), computer vision and graphics, and high-performance computing devices, accurate vehicle dynamics, photorealistic rendering, and realistic sensor simulation are now being realized and accessible. Some well-known simulators include Intel's CARLA (Dosovitskiy et al., 2017), Google/Waymo's CarCraft (Madrigal, 2017) and SimulationCity (Waymo LLC, 2021a), Tesla's simulator (Tesla, Inc., 2021), Microsoft's AirSim (Shah et al., 2018), NVIDIA's DRIVE Sim (Nvidia Corporation, 2021), Baidu's AADS (Li et al., 2019b), and Cruise's simulator (Cruise LLC, 2020), etc. Despite the above efforts and advancements, these simulators mainly focus on the fidelity of the vehicle rather than the driving environment, especially for the background road user behavior. The behaviors of background agents are either replayed from logged data or simulated using oversimplified heuristic rules, which leaves a significant gap between the simulation and the real-world driving environment.

1.2.2 Statistical realism of naturalistic driving environment

AV-related applications bring new requirements on NDE. For simulation to be an effective tool, statistical realism of NDE is a must. The simulation should reproduce the encounters that an AV

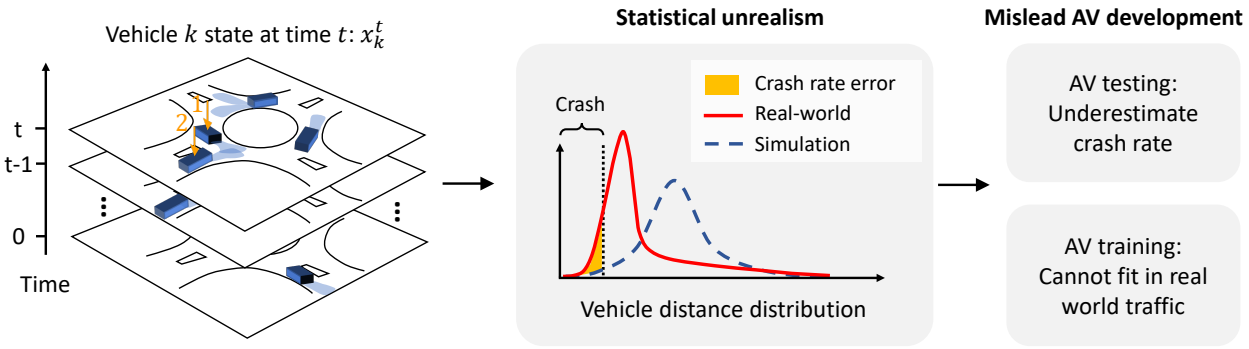


Figure 1.1: Statistical errors in simulation may mislead AV development

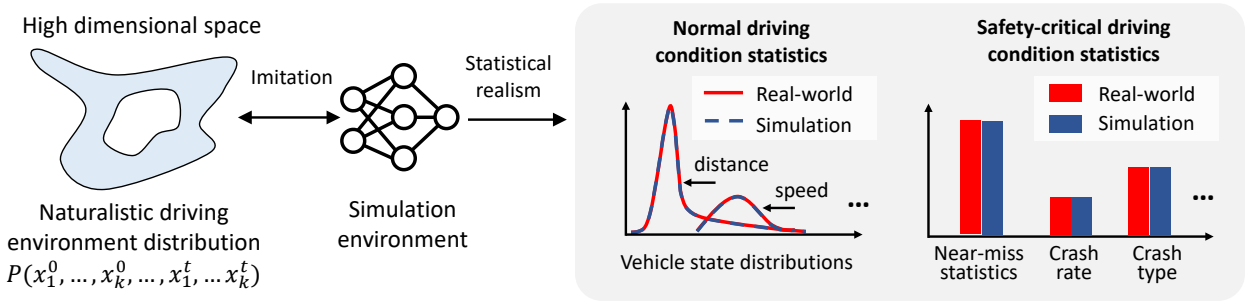


Figure 1.2: Modeling NDE with statistical realism

might face in real-world traffic with distribution-level accuracy, especially for those safety-critical events. AV applications require finer modeling accuracy than microscopic traffic simulators used in the field of transportation engineering, since nuanced differences in human driving behavior can significantly affect AV safety performance. Consequently, the calibration targets, for example, link flow and travel time, specified previously in the Federal Highway Administration guidelines (Dowling et al., 2004) are not sufficient. Unfortunately, the real-world NDE is spatiotemporally complex and highly interactive. Therefore, how to achieve statistical realism for such simulators is a long-standing problem in the field.

The lack of statistical realism for simulation can potentially mislead AV development in both training and testing. An illustration example is shown in Figure 1.1. Consider a roundabout environment that includes multiple vehicles. At time t, a vehicle (vehicle 1) is circulating, and another vehicle (vehicle 2) is about to enter the roundabout. Their potential future positions are denoted by shaded blue areas, and they have a probability to collide if vehicle 2 fails to yield. Assume the distance between the two vehicles in the real world follows certain distribution as shown by the red curve and the simulated results are the dashed blue curve. This statistical difference, i.e., distribution inconsistency between the real world and simulation, will lead to an underestimation of vehicle crash rate and therefore provide optimistic estimates of AV safety

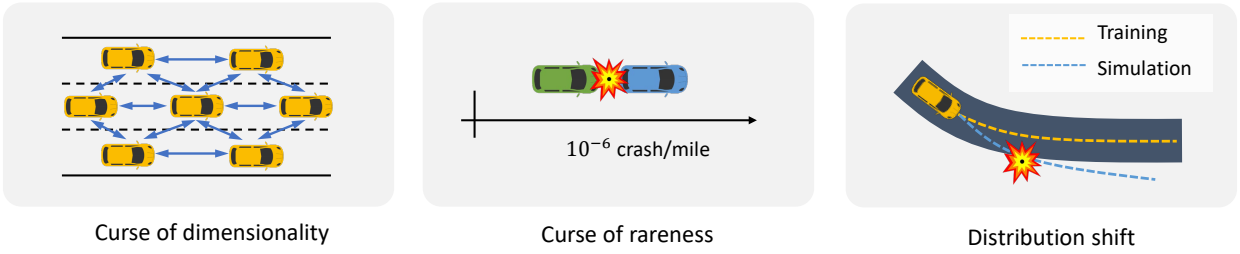


Figure 1.3: Major challenges for modeling NDE

performance. Also, since the distance between vehicles in the simulation environment is not consistent with the real world, an AV agent trained in it might not fit in real traffic due to the large sim-to-real gap. In real-world driving environments, instead of two agents, multiple human drivers are continuously interacting with each other and their states are progressively evolving for a long time horizon. Therefore, the underlying joint AV distribution is extremely complex and in a very high-dimensional space as shown in Figure 1.2. The goal of NDE modeling is to achieve distribution-level accuracy under both normal driving and safety-critical situations. Therefore, a wide range of environment statistics, for example, vehicle speed and distance distributions, crash rate, crash type and severity distributions, near-miss measurements, etc., need to be consistent with the real world.

1.2.3 Challenges

The challenges of modeling NDE with statistical realism mainly come from three aspects as shown in Figure 1.3. The first challenge is from the "curse of dimensionality". The real-world driving environment is highly interactive and spatiotemporally complex with large numbers of road users and long time horizons, which makes NDE modeling a very high-dimensional problem. The second challenge is from the "curse of rarity" (Liu and Feng, 2022). Since safety-critical events (e.g., crashes) rarely happen in the real-world driving environment (on average 10^{-6} crashes per driving mile for human drivers (Bureau of Transportation Statistics, 2018)), modeling such rare events in high fidelity requires an extremely high precision of the microscopic behavior. The compounding effects of the "curse of rarity" on top of the "curse of dimensionality" in the real world NDE will make it even more challenging. The third challenge is from the "distribution shift". Short-term and small modeling errors may accumulate both in space and time, which might lead to out-of-distribution behaviors like frequent offroad, unrealistic collision, or even the collapse of the entire simulation. Moreover, due to the highly interactive nature of the driving environment, unrealistic behaviors of a single agent will impact and propagate to all agents in the simulation.

1.2.4 Evaluate AV safety performance using NDE

The basic philosophy of evaluating the AV safety performance is to deploy it in the NDE, observe its performance, and make statistical comparisons to human driver performance. Such statistical comparisons, which are critical to determining the readiness of AVs to be widely deployed (Webb et al., 2020), could be quantified by safety performance statistics, such as the crash rate, which has been commonly used to measure the safety performance of human drivers. NDE testing is essentially a Monte Carlo (MC) estimation process, where the occurrence probability of the event of interest (e.g., crash) is approximated as the test mileage increases. However, events of interest like crashes are rare-event in NDE, which can cause the intolerable inefficiency issue for testing. It has been argued that hundreds of millions of miles and sometimes hundreds of billions of miles would be required to demonstrate the safety performance of AVs at the level of human-driven vehicles Kalra and Paddock (2016), which is extremely time-consuming even in simulations. This issue will be even more critical as some commentators have claimed that AVs need to be safer than human drivers with the factor ranging from 2 to 100 (Shladover and Nowakowski, 2019). Not to mention that a brand-new testing process may be required if configurations of AVs are changed. Therefore, intelligent testing algorithms based on NDE need to be designed to enhance testing efficiency without loss of evaluation accuracy.

1.3 Literature review

1.3.1 NDE modeling

Microscopic traffic simulators have been studied and developed in the transportation engineering domain for decades. The human driving behaviors are modeled through a combination of physics-driven models and hand-crafted rules, such as car-following models (Bando et al., 1995; Treiber et al., 2000), lane-changing models (Erdmann, 2015; Kesting et al., 2007), gap-acceptance models (Mahmassani and Sheffi, 1981), etc. Some well-known traffic simulators are Simulation of Urban MObility (SUMO) (Lopez et al., 2018), VISSIM (PTV GROUP, 2018), and AIMSUN (Yunex Traffic Group, 2022). Due to the limited capability of the underlying parametric models and manually encoded rules, the model fidelity is constrained. For example, the widely used Intelligent Driver Model (IDM) car-following model (Treiber et al., 2000) has only six parameters to calibrate, which cannot capture diverse human driving behaviors. Additionally, there are plenty of heuristic rules encoded in these simulators (e.g., SUMO (Lopez et al., 2018)), thus limiting their simulated trajectories. For example, vehicles are encouraged to follow road centerlines, therefore, the simulator rarely generates lane-swaying trajectories that are frequently seen in the real world. Furthermore, most of these behavioral models are deterministic or have only simple randomness,

for example, Gaussian noise (Laval et al., 2014; Treiber and Kesting, 2017), which makes them unable to characterize stochastic and multimodal human driving behavior.

Many attempts have been made by using neural networks (Liu et al., 2022; Mo et al., 2021; Wang et al., 2017; Xie et al., 2019; Zhu et al., 2018b), Markovian-based models (Chen et al., 2010), Bayesian networks (Schreier et al., 2016; Schulz et al., 2018), and game theory (Li et al., 2017, 2020), etc., to achieve better performance in modeling specific behaviors (e.g., car-following) or specific scenarios (e.g., unprotected left turn). However, statistical realism is not considered in most studies and they can hardly be generalized to model complex urban environments and highly interactive scenarios.

In recent years, human-driving behavior modeling has also attracted increasing attention from the AI community. In particular, there are many methods proposed for trajectory prediction (Gao et al., 2020; Gu et al., 2021; Liu et al., 2021; Ngiam et al., 2021; Rhinehart et al., 2019; Ścibior et al., 2021; Shi et al., 2022; Zhou et al., 2022; Huang et al., 2023), which is an important component for AV decision making. Short-term trajectory prediction also needs accurate multi-agent behavior modeling, however, as pointed out by several studies Suo et al. (2021); Xu et al. (2022), they cannot be directly applied for simulating NDE since they are brittle to distribution shift. In addition, trajectory prediction usually only concerns the prediction accuracy of a limited number of agents, while simulation concerns the modeling accuracy of the whole environment. Several approaches based on Imitation Learning (IL) and Generative Adversarial Imitation Learning (GAIL) have been investigated to construct the simulation environment (Igl et al., 2022; Bergamini et al., 2021; Kuefler et al., 2017; Bhattacharyya et al., 2018; Kamenev et al., 2021; Suo et al., 2021; Xu et al., 2022; Zhang et al., 2022a). For example, in Bergamini et al. (2021), the researchers formulated the simulation problem as a Markov Process, where the initial probability and the transition function are modeled using neural networks and learned from real-world data. To model an individual vehicle behavior, a CNN-based model is trained and then applied independently to all agents during the inference to predict their future positions. In Zhang et al. (2022a), the researchers proposed a Data-Driven Simulator (D2Sim) which involves two major steps. In the first step, a Transformer-based (Gatformer) model is developed to model the social interactions among vehicles and predict their future trajectories. In the second step, the candidate trajectories generated by the Gatformer are reshaped by the proposed Generative Adversarial Network (GAN)-based model (TSG-InfoGAN) to yield diverse and realistic traffic scenarios. However, statistical realism is hardly considered and cannot be achieved with these methods. For example, the crash rate of these simulation environments is significantly higher (by multiple magnitudes) than that of real-world traffic. Besides crash rate, there is no other environmental statistics regarding either normal or safety-critical driving conditions are considered. Moreover, these methods can only generate short-time simulation (i.e., a few seconds), which limits the

capability of full-length trip training and evaluation of AVs.

To reproduce high-fidelity interactions in NDE, there are also methods proposed based on logged data replay, real-world event reconstruction, or their variants (Li et al., 2019b; Scanlon et al., 2021; Yang et al., 2023). For example, the most widely used method in the industry is logged data replay. The basic idea is to replay safety-critical events that vehicle fleets encountered in the real world. In the simulation environment, background agents will follow their logged trajectories, and the AV will be controlled by the updated version of algorithms to evaluate whether it can achieve better performance. The deficiency of this method is obvious, where the background agents are non-reactive and cannot interact with the updated AV model. This means we cannot test the consequences of AV-updated behaviors, which will influence BVs behavior. Yang et al. (2023) proposed a method (i.e., UniSim) to edit logged data and generate new scenarios based on them. UniSim can generate multi-sensor data for different views and different sensor configurations. Especially, background agents can be added or removed from the recorded log to create new scenarios. However, it mainly focused on sensor simulation and the fidelity of background agents' behavior can be limited. Scanlon et al. (2021) proposed to build NDE based on real-world fatal collision events from various data sources including police reports and videos. However, it may be difficult to reconstruct near-miss events using this method since the information needed for reconstruction is usually not available.

1.3.2 AV testing

One important application of the high-fidelity NDE is to test the safety performance of AV. The prevailing approach is to drive AVs in NDE and make statistical comparisons of their performance to human drivers (Kalra and Paddock, 2016). As discussed previously, due to the rarity of safety-critical events in NDE, significant testing miles would be required to demonstrate the safety performance of AVs at the level of human-driven vehicles. To address the inefficiency issue, scenario-based approaches have been proposed (Riedmaier et al., 2020; Ma et al., 2022). The basic idea is to test AVs in purposely generated scenarios that are more safety-critical (Zhao et al., 2016; O'Kelly et al., 2018; Feng et al., 2020d,a; Li et al., 2019a; Feng et al., 2020c). Based on Importance Sampling (IS) theory, the testing efficiency can be improved since the probability of AV encountering safety-critical events can be greatly increased.

However, most existing scenario-based methods can only be applied to short scenario segments (e.g., a few seconds) of NDE with limited dynamic objects (e.g., one or two background road users). The decision variables needed to represent the scenarios are low-dimensional (Feng et al., 2021), for example, a cut-in event performed by a Background Vehicle (BV) (i.e., human driver) that lasts for a few seconds. They cannot represent the full complexity and variability of the

NDE, where the decision variables are high dimensional, particularly considering the complex interactions among BVs and AVs. For example, (O'Kelly et al., 2018) attempted to employ the scenario-based approach in relatively high-dimensional scenarios, however, it can only accelerate the testing process by 2-20 times when considering five BV. If more BVs and longer time horizon were considered, the acceleration ratio could be further reduced. However, the AV driving task in an urban area for an extended time period can be very complex, which involves interactions with hundreds of vehicles, involving various maneuvers such as car-following, lane-changing, and merging, and passing through different roadways such as roundabouts and intersections. Consequently, existing scenario-based approaches may not be able to handle them.

Moreover, testing AV through isolated scenarios, rather than within the spatially and temporally continuous NDE, may not be sufficient to provide a comprehensive assessment of AV performance. For instance, an AV that performs well in two separate scenario segments might falter when subjected to continuous testing involving the combination of these segments, due to factors like accumulated errors of AVs. This has also been pointed out by Waymo: "while short-form simulation allows us to optimize and refine isolated skills, such as nudging around a double-parked vehicle, it is important to evaluate how an autonomously driven system's capabilities work together over full-length trips" (Waymo LLC, 2021a). In their blog, they also emphasize the importance of high-fidelity NDE, and propose to evaluate AV trip-level safety performance leveraging simulations.

1.4 Dissertation overview

1.4.1 Research scope

This dissertation aims at providing systematic and generic methods for simulating high-fidelity NDE for AV development and testing, leveraging large-scale NDD. We identify and study the new and challenging requirement on NDE modeling, i.e., statistical realism, brought by AV-related applications. The simulated NDE should achieve distribution-level accuracy in both normal and safety-critical driving conditions to reproduce all potential encounters that an AV might face in real-world traffic. Only with such property can the simulation results be trustworthy. The key to NDE is modeling human driving behavior, as illustrated in Figure 1.4. We first proposed a data-driven-based method to characterize vehicle car-following and lane-changing behavior distributions, which can ensure the statistical realism of NDE in highway driving environments. To improve the modeling accuracy in more complex urban driving environments and safety-critical conditions, we further developed a deep learning-based method to jointly model multi-agent behavior, which can better learn interactions happen between multiple human drivers. The

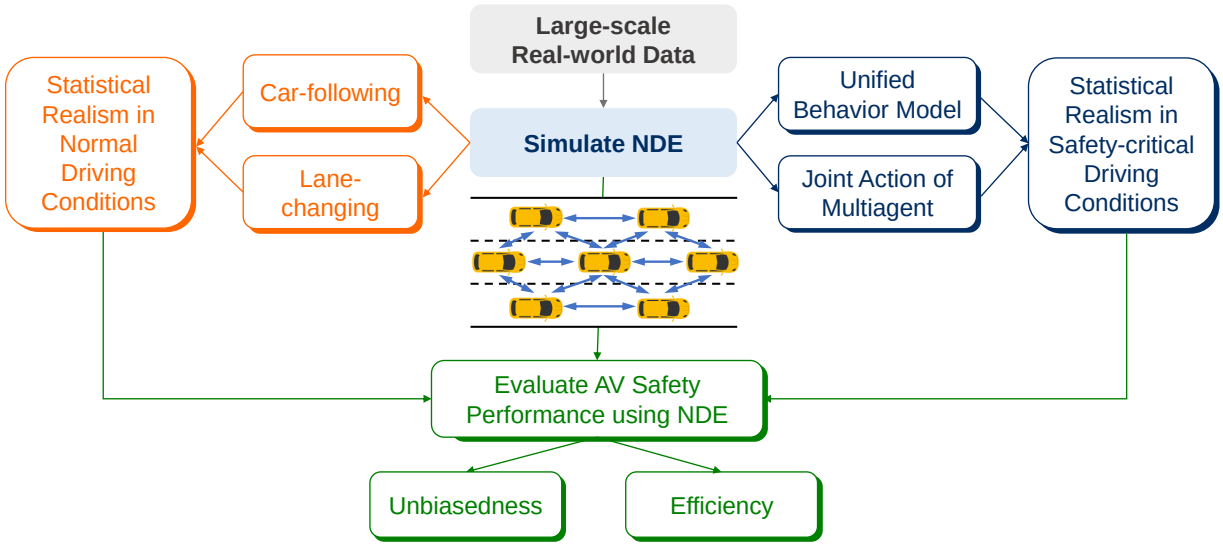


Figure 1.4: Simulating Naturalistic Driving Environment for AVs

proposed method can achieve accurate safety-critical statistics including both crashes and near-misses, which are critical for training and testing AVs. The scalability potential of the proposed method is demonstrated for simulating road networks, which enables trip-level AV training and testing. Note that an underlying assumption we apply in this dissertation is that the presence of an AV has a negligible effect on NDE. More research is needed to consider potential behavioral differences in human-driven vehicles when interacting with AV. More discussions on this can be found in Chapter 5.

One important application using high-fidelity NDE is to evaluate AV safety performance. For simulation-based AV testing, there are two critical problems that need to be solved, as shown in Figure 1.5. The first one is how to build a high-fidelity NDE based on NDD. Only with a high-fidelity NDE, the simulation results obtained from it can be unbiased. The second one is how to develop testing methodologies that can improve AV testing efficiency. Therefore, this dissertation also develops intelligent testing environment for AV safety assessment. We found that sparse but adversarial adjustments to NDE can significantly reduce the required test miles (with multiple orders of magnitude) without loss of evaluation unbiasedness. Consequently, a systematic framework for AV safety performance evaluation is developed in this dissertation, enabling trustworthy and efficient evaluation procedures.

1.4.2 Contributions

The contributions of this dissertation are listed as the following:

1. We identify and study the statistical realism requirement on NDE, a new requirement for

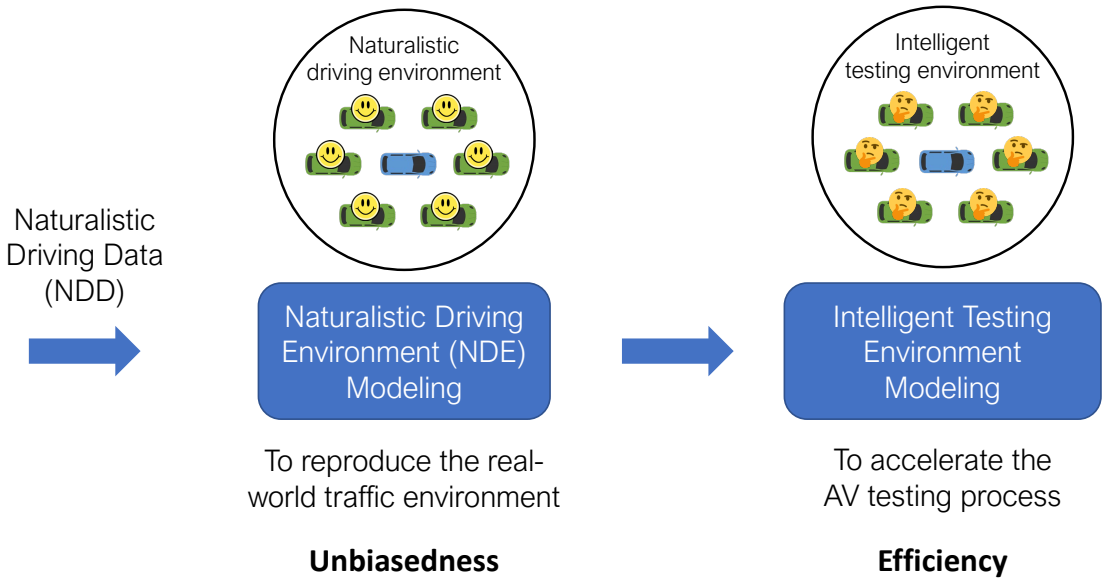


Figure 1.5: Evaluate AV safety performance unbiasedly and efficiently

microscopic traffic simulators brought by AV applications.

2. We propose a data-driven NDE modeling method, which can ensure statistical realism in long-time horizon simulations. The proposed method is validated using large-scale real-world data on highway driving environments.

3. We propose a deep learning-based NDE modeling method, which can further improve the simulation accuracy, especially for safety-critical driving statistics. The proposed method is validated using large-scale real-world data on urban driving environments.

4. We propose an intelligent testing method to evaluate AV based on NDE, which can significantly improve the testing efficiency without loss of evaluation unbiasedness.

In summary, this dissertation provides comprehensive methods for high-fidelity NDE simulation and also methodologies to accelerate the AV testing process based on NDE.

1.4.3 Overview and organization of the dissertation

Other than this introduction Chapter 1, this dissertation is organized as follows.

In Chapter 2, we propose a data-driven NDE model that can achieve statistical realism in simulating highway driving environments. Specifically, car-following and lane-changing models are developed utilizing large-scale NDD, where empirical distributions of human driving behaviors, for example, acceleration distribution and lane-changing probability, are obtained to

model stochastic human driving behaviors. Although empirical distributions are a good estimation of human driving behaviors, there are inevitable model errors due to data quantity and quality, which will accumulate and amplify during the simulation and lead to an unrealistic environment. To satisfy the statistical realism requirement, an optimization-based method is further designed to refine empirical behavior models. Specifically, the vehicle state evolution (e.g., vehicle speed) is modeled as a Markov chain and its stationary distribution is twisted to match the ground truth observed from real-world traffic. The proposed method is validated in the case study of a multi-lane highway driving simulation.

In Chapter 3, we propose a deep learning-based NDE model that can achieve statistical realism in simulating urban driving environments. Compared with highway environments, urban environments usually involve more complex interactions between multiple agents, and driving behaviors are difficult to heuristically divide into categories such as car following and lane changing. To capture the high-dimensional interaction, we frame the simulation modeling under an IL paradigm with deep neural networks under the supervision of large-scale real-world demonstration. Specifically, a Transformer-based behavior modeling network is proposed to learn multi-agent interaction behavior, and a conflict critic model and a safety mapping network are designed to control the generation process of safety-critical events, following real-world occurring frequencies and patterns. The results show that the proposed model can achieve accurate normal driving statistics (e.g., vehicle speed/distance/yielding behavior distributions, etc.), as demonstrated in the simulation of urban driving environments. More importantly, it can also reproduce accurate long-tailed safety-critical events statistics (e.g., crash rate/type/severity and near-miss statistics, etc.), which are very difficult to achieve but will notably influence AV training and testing accuracy.

In Chapter 4, we propose an intelligent testing algorithm to evaluate AV safety performance, which is an important application of high-fidelity NDE. To achieve evaluation efficiency without loss of accuracy, our approach is based on NDE, but with sparse but intelligent adjustments. The resulting driving environment is both naturalistic and adversarial, in that most of the BVs (more generally, road users) follow naturalistic behaviors for most of the time, and only at selected moments, selected vehicles execute specific designed adversarial moves. By training the BVs to learn when to execute what adversarial maneuver, the proposed environment becomes an intelligent environment for AV driving intelligence testing. We demonstrate the effectiveness of the proposed method in previously developed highway driving NDE. Compared with directly testing AV in NDE, the proposed intelligent testing algorithm can accelerate the evaluation process by multiple orders of magnitude without loss of unbiasedness.

In Chapter 5, we provide concluding remarks and discuss future research directions.

CHAPTER 2

Data-driven NDE Model

2.1 Introduction

2.1.1 Background and related works

The key to developing and testing AVs using simulation is the trustworthiness of the simulation results. As pointed out in various domains (James et al., 2019; Li et al., 2019b; Peng et al., 2018), the sim-to-real gap could hinder and even mislead the training and testing process of an agent. However, how to model the naturalistic behavior of human-driven vehicles with high fidelity still remains an open question. Many AV companies tried to replay human driving behaviors according to the logged data collected from the real-world driving environment. However, as the human driving behaviors are pre-determined in the logged data, they cannot interact with AV models, which severely limits the scenarios that can be simulated. To address this issue, the human driving models developed in the transportation engineering field have been applied, such as the IDM (Treiber et al., 2000) and Minimizing Overall Braking Induced by Lane Change (MOBIL) (Kesting et al., 2007) models. However, although these models can interact with AV models, they were designed for traffic flow analysis purposes such as reproducing traffic oscillations and the fundamental diagram, which are not suitable for the AV-related applications.

Simulating NDE for AVs brings brand new requirements on the NDE fidelity, and we will use AV testing as an illustration example to demonstrate that. Consider a car-following scenario as shown in the left side of Figure 2.1. We assume that the distance between the two vehicles in the real-world driving environment one step later follows the red-curve distribution, as shown in the middle of Figure 2.1. If the simulation is inaccurate, the simulated distance between the two vehicles will follow a different distribution, say, the dashed blue curve in the middle of Figure 2.1. This distributional discrepancy will be compounded and amplified during the simulation process, which could result in a significant difference, as shown in the right of Figure 2.1. This inconsistency can cause sim-to-real gaps, such as a lower simulated crash rate than the real-world

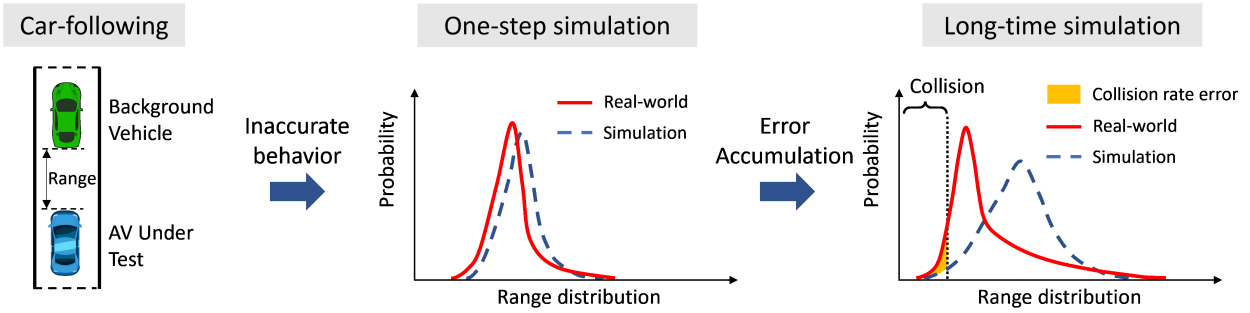


Figure 2.1: Illustration of the distribution inconsistency that might mislead AV development and deployment

crash rate. If the following vehicle is an AV, this optimistic estimation of the AV safety performance might be misleading and result in misguided AV development.

In this paragraph, we will mathematically demonstrate the statistical realism requirement on NDE, still taking AV testing as an example. To evaluate AV's safety performance quantitatively, the accident rate of AV's in NDE is usually utilized (Feng et al., 2020d, 2021, 2023; Zhao et al., 2016). Specifically, let X denote the variables that define the NDE environment. Then, testing an AV in NDE is essential to sample X from its underlying distribution, denoted as $X \sim P(X)$, and estimate its performance μ_E^κ by

$$\mu_E^\kappa := \mathbb{E}_X\left(\phi_E^\kappa\left(X\right)\right) \approx \frac{1}{n}\sum_{i=1}^{n}\phi_E^\kappa\left(X_i\right) \approx \frac{m}{n}, X_i \sim P\left(X\right), \tag{2.1}$$

where E denotes the event of interest (e.g., crash), κ denotes the AV agent under test, $\phi_E^\kappa\left(X\right)$ denotes the AV performance at the environment specified by X, n denotes the number of tests, and m denotes the number of event E occurred during tests. This equation represents the crash rate calculation of an AV using the MC simulation method, i.e., let the AV run in the NDE simulation and the crash rate is calculated by dividing the number of crashes by the total number of test runs. The MC method (Owen, 2013) requires a statistically accurate simulation environment to obtain an unbiased estimation result. Therefore, the AV simulation testing requires distributionally consistent NDE models, which are significantly different from those for traffic flow analysis. We believe the statistical realism requirement holds for many other AV applications.

To generate NDE, the common practice is to model human driving behavior using a combination of car-following and lane-changing models to control the longitudinal and lateral behavior, respectively. A general form of these models can be expressed by

$$u\left(t\right) = \psi\left(S\left(t\right), \theta\left(t\right)\right) + \epsilon\left(t\right), \tag{2.2}$$

where t denotes the time, $u\left(t\right)$ denotes the action (e.g., longitudinal acceleration or lane-change

decision) of the vehicle at the t-th time instance, $S(t)$ denotes the states of the ego-vehicle and surrounding vehicles that have the influence on the ego-vehicle's decision making, $\psi(\cdot)$ denotes the model that maps from the state space to the action space, $\theta(t)$ denotes model parameters that could be deterministic or stochastic, and $\epsilon(t)$ denotes the additive noise term.

Most traditional models (Kesting et al., 2007; Treiber et al., 2000) are deterministic and cannot capture the stochastic nature of human driving behaviors. Recently, increasing studies have used machine learning-based methods to fit the behavior model ψ using neural networks (Wang et al., 2017; Xie et al., 2019; Zhu et al., 2018b). By utilizing large-scale NDD, these methods aim to better reproduce the observed trajectories of human drivers. However, the problem of lacking accurate stochasticity still remains unsolved.

The stochasticity can be incorporated into the model by introducing external noise term $\epsilon(t)$. For external noise $\epsilon(t)$, the most commonly used one is the Gaussian noise (Laval et al., 2014; Treiber and Kesting, 2017). However, the external addition of the Gaussian noise cannot realistically depict human driving behaviors since the interaction in different driving conditions is highly complex and does not always follow the simple symmetric Gaussian distribution (Li and Chen, 2017; Yeo, 2008).

Besides adding an external noise $\epsilon(t)$, the stochasticity can also be incorporated into model parameters $\theta(t)$ (Hamdar et al., 2015; Talebpour et al., 2015; Treiber and Kesting, 2017; Yang and Peng, 2010). However, these methods do not concern with the distributional accuracy of the driving behaviors and the environment generated after the stochasticity is introduced. There were two notable exceptions in (Chen et al., 2010; Wang et al., 2009), the authors proposed stochastic car-following models to capture the distribution of time headway. They validate that the headway distribution of the simulation environment is consistent with the real world. However, only car-following behavior is considered in these studies, so they can only simulate the single-lane road. More importantly, the error accumulation issue has not been considered, which could severely distort the distribution for long-term simulation of NDE.

2.1.2 Overview of the chapter

To tackle the statistical realism requirement on NDE, this chapter aims to propose a data-driven NDE modeling method. The overall pipeline of the framework includes two major steps as shown in Figure 2.2. To accurately capture human stochastic driving behavior, in the first data-driven step, we propose to estimate the car-following acceleration distributions and lane-changing probabilities using empirical histograms obtained directly from large-scale NDD. Although empirical distributions are good estimations, the inevitable model errors caused by data quantity and quality will accumulate during the simulation and lead to inaccurate results. To tackle this

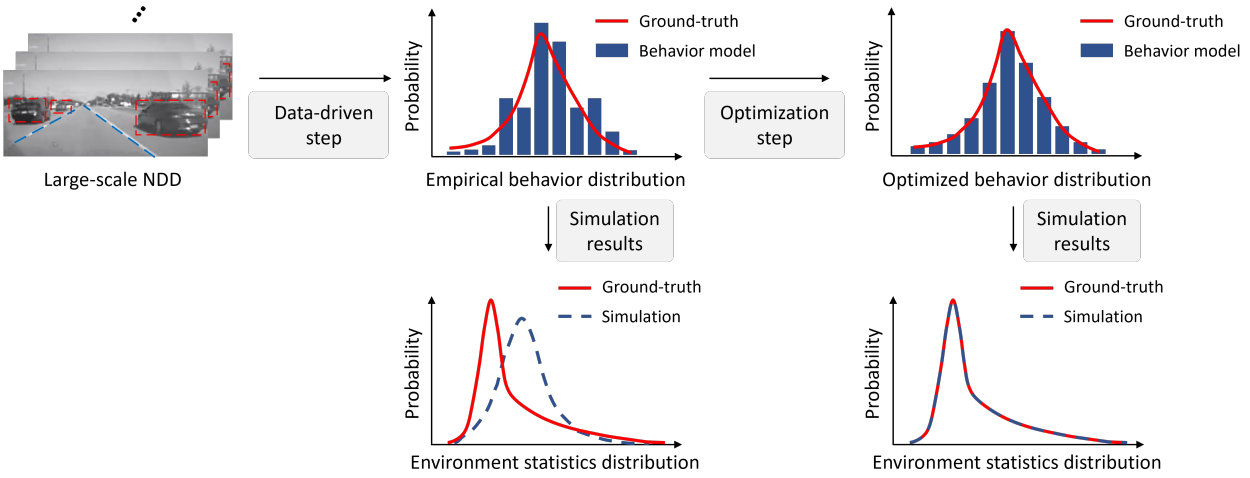


Figure 2.2: Proposed data-driven NDE modeling framework pipeline

issue, the second step of the framework is to refine empirical behavior models. To achieve that, the key is to model the long-term effects of the error accumulation. By modeling the vehicle state evolution as a Markov chain, the long-term effects of the error accumulation could be characterized by the stationary distribution of the Markov chain. Then, an optimization problem can be formulated to fit the simulated vehicle state stationary distribution with ground truth by refining the empirical behavior models. Therefore, with the optimized human behavior models, the simulated NDE can achieve accurate environment statistics (e.g., vehicle speed and distance).

2.1.3 Contributions and organization of the chapter

The main contributions of this chapter are threefold:

1. The new requirement, i.e., statistical realism, on NDE simulation for AVs are identified, which cannot be satisfied by most existing methods.

2. A novel modeling framework is proposed to generate the NDE that is statistically consistent with the real-world driving environment.

3. The proposed method is validated using large-scale real-world NDD and the generated NDE is further validated by testing AV models in Chapter 4.

This chapter is organized as follows: In Section 2.2, we introduce the data-driven step to construct empirical behavior models using large-scale real-world NDD and evaluate their performances in a multilane highway driving environment simulation. In Section 2.3, to account for the error accumulation problem, the optimization modeling step is proposed to optimize

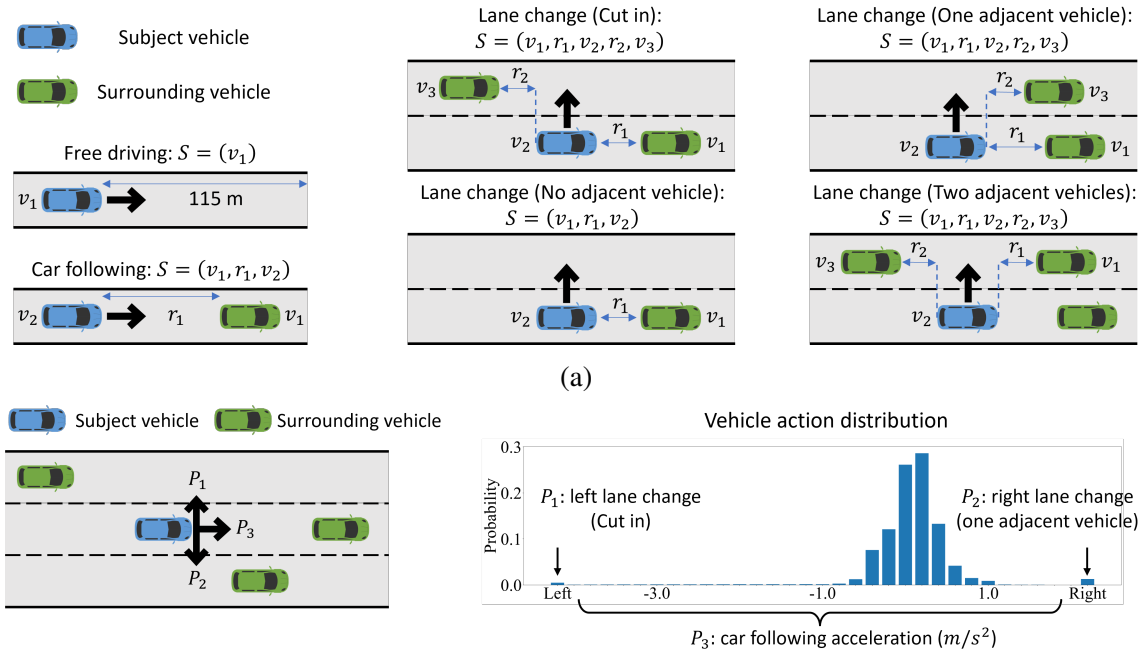


Figure 2.3: Illustration of empirical behavior models. (a) Longitudinal and lateral behavior models. (b) Illustration example of how to simulate a vehicle's action at a decision moment.

the empirical behavior models to obtain a distributionally consistent NDE. In Section 2.4, the performance of the proposed method is validated and compared with existing models. We conclude this chapter in Section 2.5.

2.2 Data-driven modeling step

In this section, we propose a simple yet effective data-driven method for NDE modeling leveraging large-scale NDD. Specifically, six empirical behavior models are constructed including free-driving, car-following, and four lane-changing behaviors with different driving conditions, and then the NDE can be generated by combining the six empirical behavior models according to the driving condition at each time step (Section 2.2.1). To construct each empirical behavior model, the large-scale NDD is processed and utilized in Section 2.2.2. Then, a multi-lane highway driving environment is simulated to evaluate the performance of the empirical behavior models in Section 2.2.3, which validates the data-driven method and further motivates the optimization modeling step in Section 2.3.

2.2.1 Empirical behavior models

To construct the NDE, both longitudinal and lateral behaviors of human drivers need to be modeled based on the vehicle's own state and its surrounding situations S. In this study, six behavior models are proposed including free-driving, car-following, and four lane-changing behaviors with different driving conditions, as shown in Figure 2.3a. Specifically, the vehicle acceleration in the free-driving case is modeled that depends only on its current velocity, while the acceleration in the car-following case is modeled that depends on the velocity, range (relative position), and range rate (relative speed) of the subject vehicle and its preceding vehicle. To capture the lane-changing probability in different conditions, four lane-changing models are proposed, which output the lane change probability of the subject vehicle at each moment. For example, in the cut-in lane change situation, the lane-changing probability depends on velocities and distances between the subject vehicle and the preceding vehicle in the current lane and the vehicle behind in the target lane. We note that more lane-changing models could be constructed in this framework by dividing the driving conditions into more categories if needed.

After constructing the six behavior models, the NDE can be generated by combining the behavior models according to the driving condition at each time step. Taking Figure 2.3b (left) as an example, the subject vehicle can take left lane change, keep car-following, or take right lane change at the moment. The left lane change behavior can be categorized as a cut-in behavior, where the lane-changing probability P_1 can be obtained from the cut-in behavior model. The right lane change behavior can be categorized as a lane change with one adjacent vehicle, where the lane-changing probability P_2 can also be obtained by the corresponding behavior model. Moreover, the longitudinal acceleration probability P_3 can be obtained by the car-following behavior model. After normalization, we can obtain the action distribution of the vehicle as shown in Figure 2.3b (right). Then, the subject vehicle's action will be sampled from this distribution and used to update its state to the next time step. To simplify the modeling process, longitudinal acceleration is assumed zero if the vehicle is making a lane change behavior. Also, if there is no vehicle in front, the ego-vehicle will not take lane-changing behavior. By repeating this process for all vehicles and time steps, the NDE can be generated.

The remaining question is how to construct the six behavior models with distributional accuracy. In this chapter, we propose to directly estimate empirical behavior distributions as the behavior models by leveraging large-scale NDD. As the NDD records all the information needed for human driving behaviors, accurate empirical behavior models could be constructed if using a sufficient amount of data with perfect quality. Although the actual data is usually limited by the data quality and quantity, these empirical behavior models could provide a good foundation and can be further improved as discussed in Section 2.3. Specifically, for each behavior model, we obtain the empirical probability $P(a|S)$ for all the vehicle actions $a \in \mathcal{A}$ at all

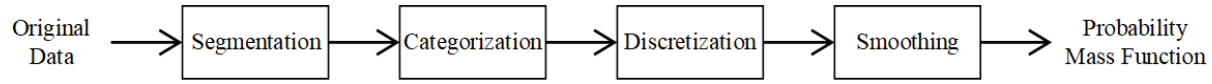


Figure 2.4: Data processing flow chart

discretized states $S \in \mathcal{S}$, where $\mathcal{A}$ denotes the action space, and $\mathcal{S}$ denotes the state space. Let $F(S) = \left[P(a_1|S), \cdots F\left(a_{|\mathcal{A}|}|S\right)\right]$ denote the probability mass function under a certain state S, then the empirical behavior model can be denoted by

$$F = \left[F(S_1), \cdots F\left(S_{|\mathcal{S}|}\right)\right] \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}. \tag{2.3}$$

Next, we will introduce how to process the NDD and construct F for all the six behavior models.

2.2.2 Naturalistic driving dataset processing

To construct empirical behavior models, we utilized large-scale NDD from the Integrated Vehicle Based Safety System (IVBSS) dataset (Sayer et al., 2011) and the Safety Pilot Model Deployment (SPMD) dataset (Bezzina and Sayer, 2014) at the University of Michigan Transportation Research Institute (UMTRI). In the IVBSS program, 108 drivers ranging from 20 to 70 years old were recruited. Each participant drove the IVBSS vehicle equipped with the Data Acquisition System (DAS) for 6 weeks. The relative distance and speed with the leading vehicle are recorded by radar at 10 Hz. The SPMD program covered over 34.9 million travel miles and included 98 vehicles equipped with the DAS and Mobileye to record human naturalistic driving behaviors. The data were also recorded at 10 Hz with positions, speeds, and accelerations of ego-vehicles, relative speeds with surrounding vehicles, and both longitudinal and lateral distances between vehicles and lane markings. We queried partial datasets with the following criteria: (1) vehicle was traveling at a speed between 20 m/s and 40 m/s; (2) dry surface condition; (3) daylight condition. The resulting dataset includes approximately 8,200 driving hours data.

The data processing consists of four steps including segmentation, categorization, discretization, and smoothing, as shown in Figure 2.4. Specifically, the original data were first segmented into trajectories and then categorized into specific groups based on the six driving situations defined in empirical behavior models. Then, a smoothing technique was applied to the discretized action distribution and finally, we could obtain the probability mass functions for each group, which constituted the six empirical behavior models. More details of the data processing steps can be found in Appendix A.1. Figure 2.5 demonstrates examples of constructed empirical behavior models. Specifically, Figure 2.5a and Figure 2.5b show examples of the vehicle longitudinal acceleration distributions in free-driving and car-following situations, respectively.

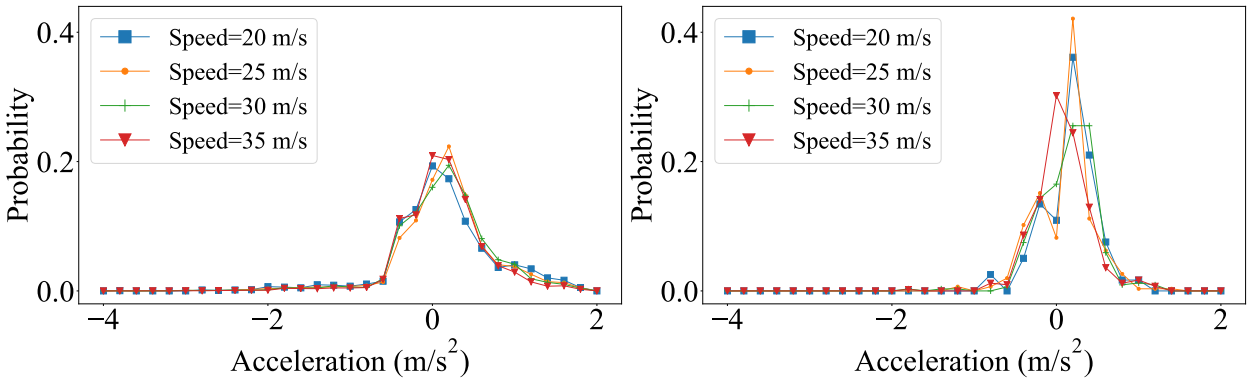


Figure 2.5: Examples of empirical behavior models. (a) Free driving, (b) Car following ($r_1 = 30\ m, v_1 = v_2$).

For the car-following case, Figure 2.5b indicates acceleration distribution when the ego-vehicle and its preceding vehicle have the same speed and their range is 30 meters. We can find that for both free-driving and car-following cases, the mean of acceleration is around zero, which is consistent with the intuition. Compared with the car-following situation, the probability of acceleration greater than zero is generally higher in the free-driving case, which is reasonable as well.

2.2.3 Performance evaluation of empirical behavior models

In this subsection, the performance of the NDE constructed by the six empirical behavior models is evaluated in a three-lane highway simulation, as illustrated in Figure 2.3b (left). We ran 100 simulation episodes to mitigate the randomness effect. To fully examine the error accumulation issue, each simulation ran 15 minutes, which included 10 minutes of warm-up time and 5 minutes of data collection. The detailed simulation settings can be found in Appendix A.2. The Hellinger distance (Wikipedia contributors, 2022) is used to quantitatively measure the dissimilarity between the simulated distribution and the true distribution. The Hellinger distance ranges from 0 to 1, and the smaller the measurement, the better the model performance. To demonstrate the performance, the BVs velocity and range distributions, which are important for the AV testing, are investigated as shown in Figure 2.6. The yellow bars show the simulation results and the blue bars show the ground-truth distributions. Results show that although the distributions can roughly capture the trends of the real-world distributions, there still exists significant distributional inconsistency, particularly for vehicle velocity. This inconsistency is caused by the error accumulation of the empirical behavior models, where the small model errors are accumulated and amplified along with the simulation steps, as illustrated in Figure 1.1. To address this issue, the optimization

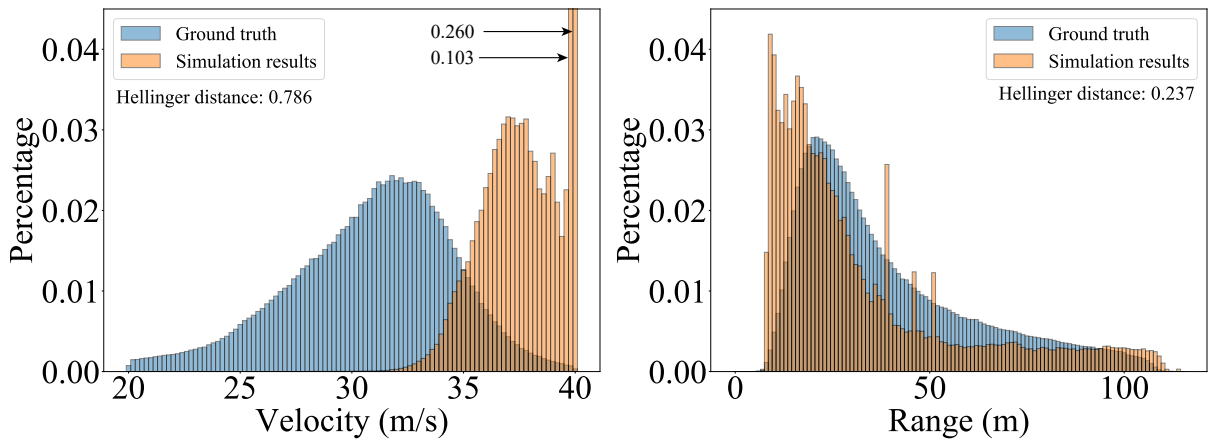


Figure 2.6: Velocity and range distributions of the NDE using empirical behavior models.

modeling step is developed in the next section to further improve the empirical models.

2.3 Optimization modeling step

In this section, the optimization step is proposed to refine empirical behavior models to minimize the accumulated errors using optimization methods. To achieve this goal, the key is to model the long-term effects of the error accumulation. Specifically, by modeling the vehicle state evolution as a Markov chain, the long-term effects of the error accumulation can be characterized by the stationary distribution of the Markov chain. Then, an optimization problem is formulated to minimize the accumulated errors by adjusting the empirical behavior models, which results in the NDE model with more accurate distributions. In the following paragraphs, we first propose the optimization framework in Section 2.3.1 and then apply the framework to the longitudinal behavior models in Section 2.3.2.

2.3.1 Optimization framework

In order to solve the error accumulation problem, we need to measure the accumulated error of the NDE generated by the empirical behavior models. One possible way is to simulate the NDE, collect the data, and obtain the simulated NDE distribution. However, the computational burden of this method is very heavy since a large number of simulations are needed to obtain an accurate estimation. To overcome this problem, we propose to measure the NDE distribution by analyzing the stationary distribution of the NDE Markov chain. By using this analytical method, the NDE stationary distribution serves as an accurate approximation of the simulated environment, which reflects the performance of the empirical behavior models. By fitting the simulated stationary

distribution with real-world ground-truth, we can improve the NDE accuracy and solve the error accumulation issue.

$$\min_{F} \quad \text{distance}(F, F^\star)$$

$s.t.$ 1. Definition of the stationary distribution.

2. Relation between behavior model and Markov chain transition probability.

3. Relation between stationary distribution and ground-truth distribution.

4. Other constraints.

Figure 2.7: Overall formulation of the proposed optimization modeling step

Following this idea, the optimization framework can be formulated as in Figure 2.7. The decision variable is the vehicle behavior model F as shown in Equation 2.3, and the objective is to minimize the adjustment to the empirical behavior model $F^\star$ while ensuring the accuracy of the stationary distribution. To achieve this objective, there are generally four sets of constraints in the optimization formulation. The first set of constraints is the standard definition of stationary distribution, which indicates that the state will always follow its stationary distribution after reaching the steady state. The second set of constraints describes the relationship between the behavior model and Markov chain state transition probability. The stochastic vehicle behavior model outputs actions for the next time step and therefore determines the state transition process. The third set of constraints is to match the stationary distribution of the simulation with the real-world ground-truth distribution, which is the key to reducing the accumulated errors. As a result, the simulated environment can be guaranteed to fit the desired real-world statistics (e.g., velocity and range distributions) even after a long simulation time horizon. The last set of constraints denotes other standard requirements, such as non-negative constraints of probability mass functions, normalization of stationary distribution, etc.

2.3.2 Optimization of longitudinal behavior models

In this section, we apply the proposed framework to optimize the two empirical longitudinal behavior models as a proof of concept, while keeping the four empirical lateral behavior models unchanged. As the velocities cannot be well modeled by the empirical models as shown in Figure 2.6, we choose the velocity distribution as the optimization target.

For the free-driving behavior model, we define the discretized speed as the state of the Markov chain. It is easy to find that the Markov chain is finite, irreducible, and aperiodic, so there exists a

unique positive stationary distribution π (Grimmett and Stirzaker, 2020) satisfying

$$\pi^T \mathbf{P} = \pi^T, \tag{2.4}$$

$$\sum_{S \in \mathcal{S}} \pi_S = 1, \tag{2.5}$$

$$\pi \succcurlyeq 0, \tag{2.6}$$

where $\mathbf{P}$ is the state transition probability matrix. As the vehicle longitudinal acceleration depends only on its current speed in the free-driving situation, the state transition probability matrix $\mathbf{P}$ is essentially a function of the behavior model F in Equation 2.3 as

$$\mathbf{P}\left(S_i, S_j\right) = G\left(F\right), \forall S_i, S_j \in \mathcal{S}, \tag{2.7}$$

where $G\left(\cdot\right)$ is a linear mapping from the longitudinal acceleration to the state transition. For example, if the current speed falls in the state S_i, the next speed after the transition is S_j, the time resolution is Δt, and the probability of taking acceleration a that satisfies the $S_j = S_i + a \cdot \Delta t$ is $p\left(a|S_i\right)$, then $\mathbf{P}\left(S_i, S_j\right) = p\left(a|S_i\right)$. Moreover, as the goal of the optimization is to match the vehicle stationary speed distribution with the real-world speed distribution in the free-driving situation, we have

$$\pi = \pi^\star, \tag{2.8}$$

where $\pi^\star$ is the ground truth of the speed distribution in free-driving situations that is obtained from the large-scale NDD.

Finally, the optimization problem can be formulated as below:

$$\min_{F} \quad \left\|F - F^\star\right\|_{Frob} \tag{2.9}$$

$$\text{s.t.} \quad \pi^T \mathbf{P} = \pi^T, \tag{2.10}$$

$$G\left(F\right) = \mathbf{P}\left(S_i, S_j\right), \forall S_i, S_j \in \mathcal{S}, \tag{2.11}$$

$$\pi = \pi^\star, \tag{2.12}$$

$$\sum_{a \in \mathcal{A}} F\left(a|S\right) = 1, \forall S \in \mathcal{S}, \tag{2.13}$$

$$\sum_{S_j \in \mathcal{S}} \mathbf{P}\left(S_i, S_j\right) = 1, \forall S_i \in \mathcal{S}, \tag{2.14}$$

$$\sum_{S \in \mathcal{S}} \pi_S = 1, \tag{2.15}$$

$$F, \mathbf{P}, \pi \succcurlyeq 0. \tag{2.16}$$

The Frobenius norm $\left\|\cdot\right\|_{Frob}$ is adopted to measure the distance between the optimized

free-driving model F and empirical free-driving model $F^{\star}$. Compared with the constraints discussed in Figure 2.7, Equation 2.10 denotes the definition of the stationary distribution, Equation 2.11 denotes the relation between behavior model and transition probability, Equation 2.12 denotes the distributional consistency between the stationary distribution and the ground-truth, and Equations 2.13-2.16 denote other constraints including the normalization requirements for the acceleration probability mass function, state transition probability matrix, and stationary distribution, respectively, and the non-negative requirements. It can be found that this is a linear programming problem that can be solved efficiently using commercial solvers, for example, Gurobi (Gurobi Optimization, LLC, 2022).

For the car-following situation, the vehicle state is composed of the speed of the subject vehicle (v), range (r), and range rate (rr) with the preceding vehicle. The state transition in the car-following situation depends not only on the subject vehicle action but also preceding vehicle action, which makes the optimization problem more complex. To solve this issue, we optimize the steady-state situation of the car-following model, which is a necessary condition regardless of the evolving process of the stationary distribution. As the preceding vehicle has reached the steady state, the ego-vehicle state transition relies only upon its own action. Then, the optimization problem can be formulated as the same as Equations (2.9-2.16), where decision variables are the probability mass functions of the car-following accelerations (F in Equation 2.3), and Equation 2.10 is a three-dimensional joint state distribution (i.e., v, r, and rr). It is also a linear programming problem that can be solved efficiently.

2.4 Performance evaluation

In this section, the performance of the proposed NDE modeling framework in highway driving environment is evaluated. We examine whether the proposed NDE can generate accurate velocity and range distributions, compared with the existing NDE baseline (i.e., SUMO (Lopez et al., 2018)) and the empirical models constructed in Section 2.2. Specifically, three existing car-following models, the Wiedemann 99 model (W99) (VISSIM, 2012), IDM (Treiber et al., 2000) model, and the stochastic IDM model are selected as car-following model baselines, which are widely applied in existing traffic simulators. The SUMO LC2013 lane-changing model is used for lateral behavior. For fair comparisons, the model parameters are calibrated with the SPMD NDD described in Section 2.2.2 using the calibration method developed in literature (Hammit et al., 2018; Hammit, 2018). More simulation settings can be found in Appendix A.2.

Figure 2.8 shows the results of the proposed NDE model and existing NDE models. It can be found that the proposed NDE model can significantly better reproduce the real-world velocity and range distributions than existing ones. Specifically, both the IDM and Wiedemann 99 models

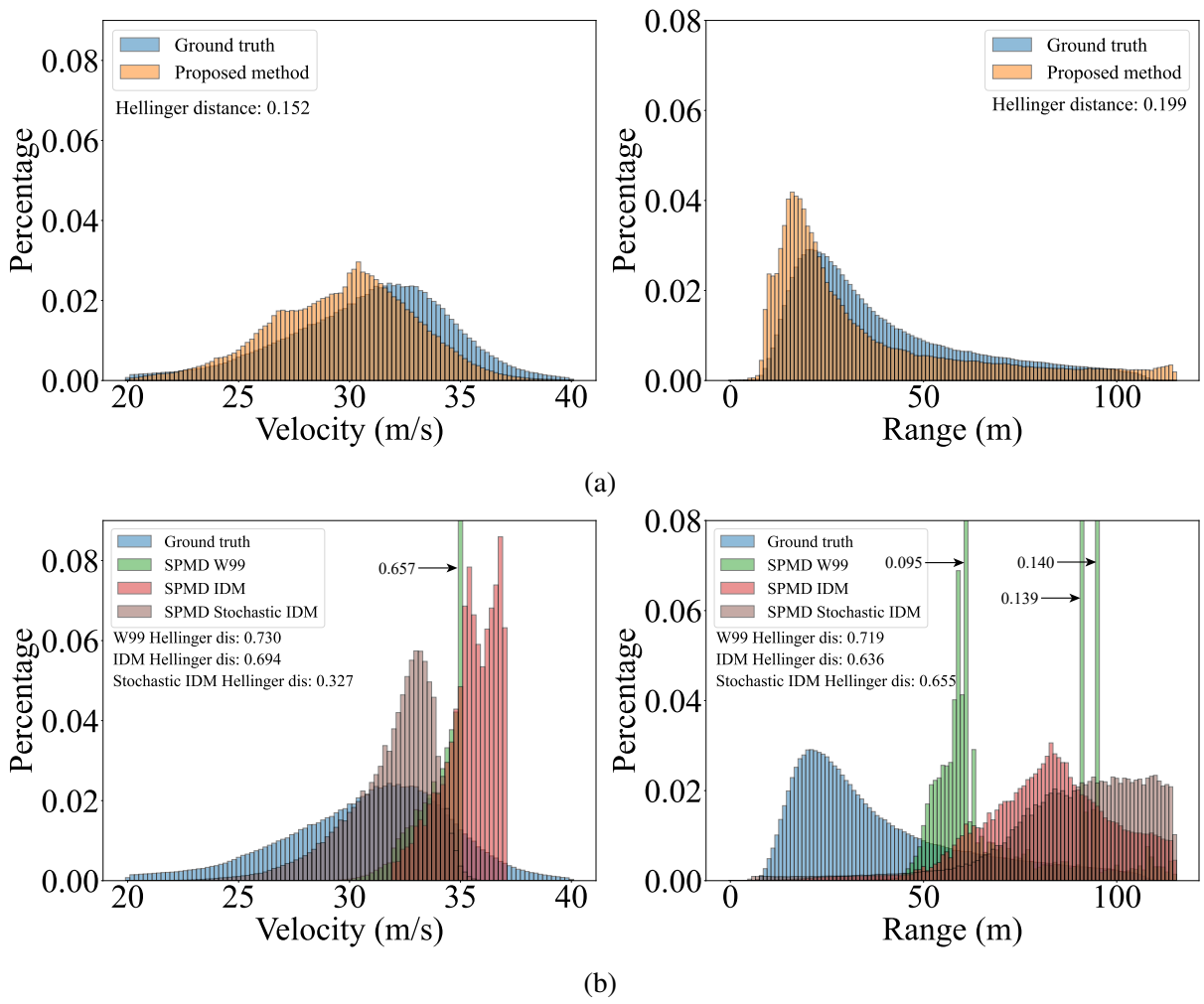


Figure 2.8: Velocity and range distributions of the. (a) Proposed method. (b) SUMO simulator.

are concentrated in a small interval of velocity and range, while the real-world distributions range among a much wider interval. The stochastic IDM model can generate better velocity results, however, the range distribution is still inaccurate. It is reasonable as these existing models are designed for accident-free purposes and therefore might be more conservative and have a large car-following distance. Besides using model parameters calibrated in this study, we also examine the performance using model parameters provided from the literature, where similar results are obtained (see Appendix A.4 for more details).

Compared with the results before the optimization as shown in Figure 2.6, the accuracy of the simulated velocity distribution is also significantly improved, which validates the effectiveness of the optimization step. The descriptive statistics, including the mean, range, and standard deviation are shown in Table. 2.1. From the results, we can find that the proposed method achieves the overall best performance, generating accurate results that align closely with the ground truth data.

Table 2.1: Descriptive statistics for different methods results.

Method \ Statistics	Velocity (m/s)					Range (m)				
	Min	Max	Mean	Range	Standard Deviation	Min	Max	Mean	Range	Standard Deviation
Empirical behavior models	23.60	40.00	38.12	16.40	1.83	4.85	114.91	32.72	110.06	24.83
SUMO (SPMD W99)	25.19	35.00	34.53	9.81	0.89	7.04	115.00	73.70	107.96	18.13
SUMO (SPMD IDM)	31.49	37.00	35.45	5.51	1.17	5.10	115.00	82.06	109.90	17.25
SUMO (SPMD stochastic IDM)	20.83	36.52	31.88	15.69	1.99	5.11	115.00	88.88	109.89	20.00
Proposed method	20.00	40.00	29.72	20.00	3.21	0.004	115.00	34.76	115.0	25.96
Ground truth	20.00	40.00	30.86	20.00	3.62	3.94	114.88	38.89	110.94	22.48

To further quantify the performance in terms of the entire distribution, the Hellinger distances of all these models are calculated as listed in Table. 2.2. Hellinger distance is a type of f-divergence commonly used in probability and statistics domains, to measure the similarity between two distributions. With a range of values from 0 to 1, a smaller Hellinger distance signifies a better goodness-of-fit. Based on the results, we can observe that the proposed method outperforms existing SUMO-based methods. Moreover, it shows a significant improvement over the empirical behavior models, which demonstrates the effectiveness of the optimization step.

Table 2.2: Quantitative performance evaluation for different methods.

Method \ **Metric**	**Velocity**	**Range**
Empirical behavior models	0.786	0.237
SUMO (SPMD IDM)	0.755	0.659
SUMO (SPMD W99)	0.629	0.691
Proposed method	**0.147**	**0.197**

In addition to the velocity and range distributions, we also calculated the lane-changing statistic of the proposed NDE to further examine its lateral behavior performance. From the simulation results, the average travel distance for one lane change is 4.86 kilometers. In the real-world driving environment, the same statistic is 4.45 kilometers per lane change on the highway (Lee et al., 2004). Therefore, the proposed NDE can also reproduce a reasonable number of lane changes as in the real-world driving environment, which can further demonstrate the fidelity of the proposed NDE.

2.5 Summary

In this chapter, we propose a data-driven NDE modeling framework. The proposed method can leverage large-scale NDD that has been widely collected in recent years with the advances in sensing and data acquisition technologies. The proposed driving behavior models are developed based on the histogram distributions of empirical data, which can accurately capture the stochasticity and are not restricted to certain parametric distributions (e.g., Gaussian distribution). To address the error accumulation issue and guarantee the accuracy of the NDE throughout the simulation, an optimization modeling step based on the Markov process is proposed, which optimizes the empirical models by matching simulated stationary distributions with the ground truth. The proposed method is validated for the multilane highway driving environment. The vehicle speed and range distributions and lane-changing statistics generated by the proposed NDE are consistent with the empirical ground truth, which is important for AV development and testing. Moreover, the generated NDE is utilized to test the safety performance of an AV agent, which further validates the effectiveness of the proposed method.

One limitation of the proposed methodology lies in the heuristic decomposition of human driving behavior into a combination of models, specifically car-following and lane-changing models, as in conventional microscopic traffic simulators (e.g., SUMO (Lopez et al., 2018), VISSIM (PTV GROUP, 2018)). This approach may present challenges when scaling to model complex urban environments because human driving behavior in such situations is intricate and resists neat decomposition into sub-models. Additionally, akin to the majority of existing studies, another constraint is that the proposed model solely relies on the present moment's state as input. Yet, it's crucial to acknowledge that human drivers are non-Markovian in nature, and their future behavior is influenced by historical states within a temporal window. Furthermore, in the context of AV development, the focus on safety-critical events supersedes that of normal driving conditions, as these events offer more valuable insights for AV training and testing. Hence, the statistical realism of safety-critical driving conditions (e.g., crash type and severity, near-miss measurements, etc.) should be further studied and validated. These limitations and considerations serve as the driving force for the subsequent chapter's research endeavors.

CHAPTER 3

Deep Learning-based NDE Model

3.1 Introduction

3.1.1 Background and related works

In the previous chapter, following the conventional paradigm, vehicle behavior is modeled through a combination of car-following and lane-changing models, which can hardly be generalized and scaled to model complex urban environments. Also, at each time step, all agents apply the behavior models independently which cannot fully capture the future interaction among agents and might not generate accurate safety-critical events. In this chapter, to tackle these issues, we propose NeuralNDE, a deep learning-based framework that can model all agents' behavior jointly and characterize both vehicle-to-vehicle interactions and their long-term state trajectories within a certain temporal range. The proposed framework is validated to be able to reproduce not only normal driving statistics, but more importantly, accurate safety-critical driving statistics, which are crucial for AV development.

Please refer to Section 1.3.1 for a more detailed literature review for NDE modeling with real-world data. Some representative works include Igl et al. (2022); Bergamini et al. (2021); Kuefler et al. (2017); Bhattacharyya et al. (2018); Kamenev et al. (2021); Suo et al. (2021); Xu et al. (2022); Zhang et al. (2022a). For example, the simulation environment proposed by Suo et al. (2021) can generate diverse scenarios, and no heuristic rules are needed for the simulation pipeline. Also, human driving behaviors in SimNet (Bergamini et al., 2021) and D2Sim (Zhang et al., 2022a) are shown to be able to achieve satisfactory performance in displacement error in short-term simulations. However, the statistical realism requirement is hardly considered and cannot be achieved by these methods. For example, the crash rate of all these methods is significantly higher (multiple magnitudes) than in the real world. Additionally, none of the existing methods validated the safety-critical statistics from simulation with real-world ground truth.

3.1.2 Overview of the chapter

The focus of this chapter is to build a high-fidelity NDE model that is statistically representative of real-world driving environments, particularly for those long-tail safety-critical events. This differentiates our proposed NeuralNDE model from existing simulators built based on real data.

Specifically, a behavior modeling network is built to learn multi-agent interaction behavior from vehicle trajectory data. It learns the joint distribution of multi-agents future actions, which can better model human drivers' interaction. Then, a conflict critic module and a safety mapping network are developed to refine the generation process of safety-critical events, following real-world occurring frequencies and patterns. These modules constitute a controllable safety-critical event generation mechanism, which is crucial for ensuring accurate safety-critical statistics.

The results show that NeuralNDE can achieve both accurate normal driving statistics (e.g., vehicle speed/distance/yielding behavior distributions, etc.) and safety-critical driving statistics (e.g., crash rate/type/ severity and near-miss statistics, etc.), as demonstrated in the simulation of urban driving environments. The fidelity of NeuralNDE-generated crash events is further validated against real-world crash videos and police crash reports. The scalability of the proposed method is also verified, demonstrating its potential for simulating large road networks.

3.1.3 Contributions and organization of the chapter

The contributions of this chapter are summarized as follows:

1. We propose a deep learning-based NDE modeling framework (i.e., NeuralNDE), which can achieve statistical realism, particularly for those long-tail safety-critical events that are critical to AV safety. The high-fidelity NeuralNDE simulator serves as the foundation for simulation-based AV applications.

2. The proposed method is validated on multiple real-world datasets, demonstrating its strong performance and robustness.

3. The proposed method has the potential to model large-scale road networks, enabling full-length trip training and evaluating of AVs.

This chapter is organized as follows: Section 3.2 introduces the overall framework of NeuralNDE and discusses the detailed methodology of the behavior modeling network, conflict critic module, safety mapping network, and generative adversarial training. Section 3.3 introduces more details of the proposed model, including datasets, network architecture, and training details. Section 3.4 is a case study demonstrating the performance of the proposed method, and Section 3.5 gives a summary of this chapter.

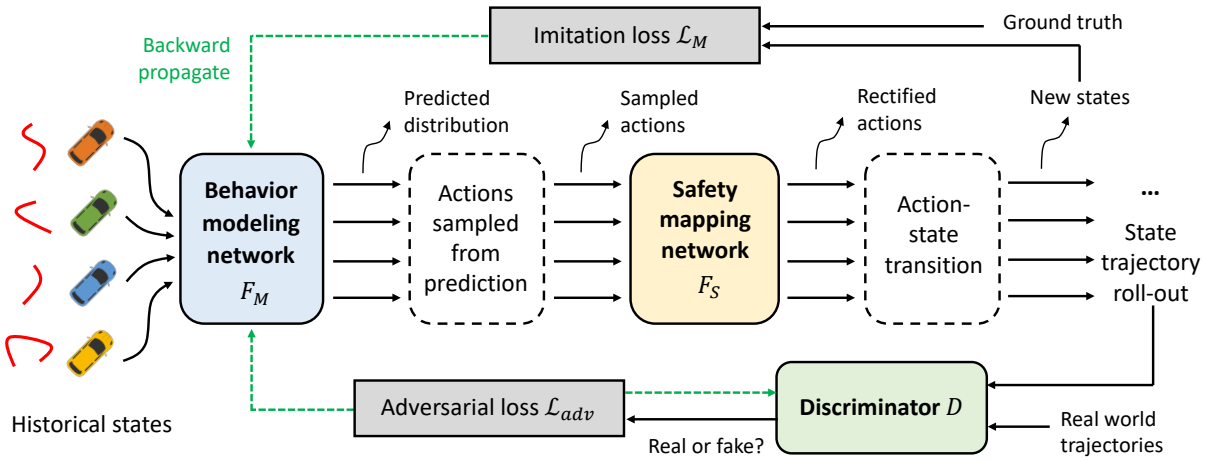


Figure 3.1: The framework and training pipeline of the NeuralNDE

3.2 Methodology

3.2.1 Overall framework

The overview of the proposed framework is shown in Figure 3.1. We frame the simulation modeling under an IL paradigm with deep neural networks under the supervision of large-scale real-world demonstration. The behavior modeling network takes in all road users' past states within a historical time window as input and predicts their joint distribution of future actions. We leverage the recent advances in fundamental models (e.g., GPT (Brown et al., 2020) and BERT (Devlin et al., 2018)) and use Transformer as the backbone of the behavior modeling network to characterize multi-agent interaction behaviors. The multi-agent actions will be sampled from the predicted distribution and passed through the safety mapping network (will be discussed in detail in Section 3.2.4) to simulate the vehicle state at future moments. To further overcome the distribution shift issue, we integrate the generative adversarial training as in GAN (Goodfellow et al., 2020) and GAIL (Ho and Ermon, 2016), where a discriminator is introduced to be jointly trained with the behavior modeling network. The simulated trajectory will be rolled out multiple times in an autoregressive manner to generate long trajectories and input into the discriminator. Therefore, two types of loss, i.e., imitation loss and adversarial loss, will be backpropagated to train the behavior modeling network to learn multi-agent interactive behaviors. The adversarial loss will also be used to train the discriminator to distinguish between real-world and simulated trajectories.

The behavior modeling network can achieve distribution-level accuracy in normal driving conditions, however, it cannot achieve such accuracy in safety-critical conditions, due to the rarity of safety-critical events in the training data, which will lead to inaccurate statistics like unrealistically high crash rates. To tackle this issue, a conflict critic mechanism is introduced

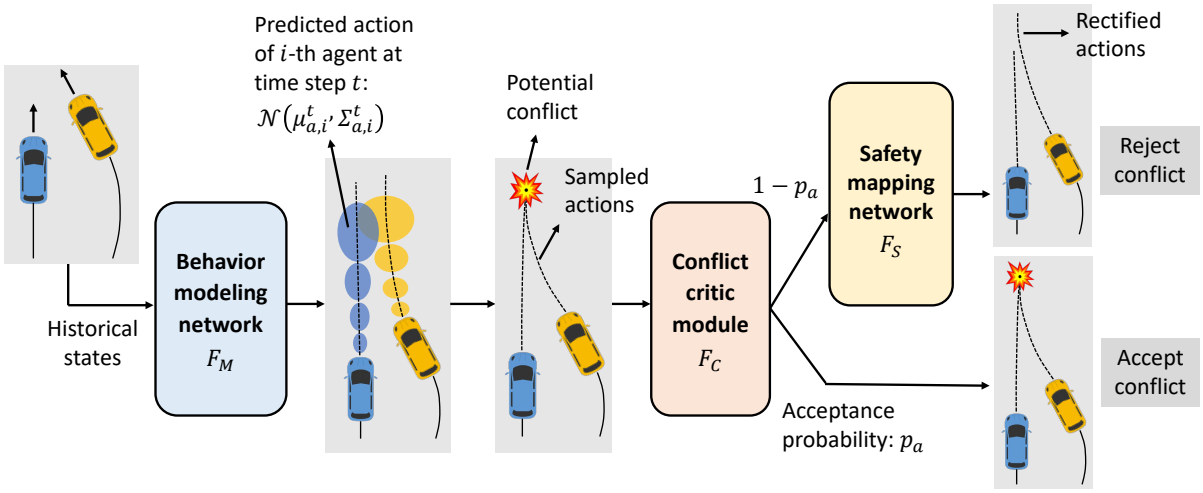


Figure 3.2: Demonstration of the behavior modeling network, conflict critic module, and safety mapping network during the inference time

during the inference time as shown in Figure 3.2. It will monitor the generated trajectories, and if there is a potential conflict, there is a certain probability of accepting vehicles performing dangerous behavior, which makes NeuralNDE capable of realizing accurate safety-critical statistics. Otherwise, the generated behaviors will be guided and rectified by the safety mapping network to resolve the conflict. The acceptance probability is trajectory-dependent and will be calibrated to fit ground-truth safety-critical statistics (e.g., crash rate and crash type distribution). Therefore, the conflict critic module controls the occurring frequencies and patterns of dangerous driving behavior during the simulation.

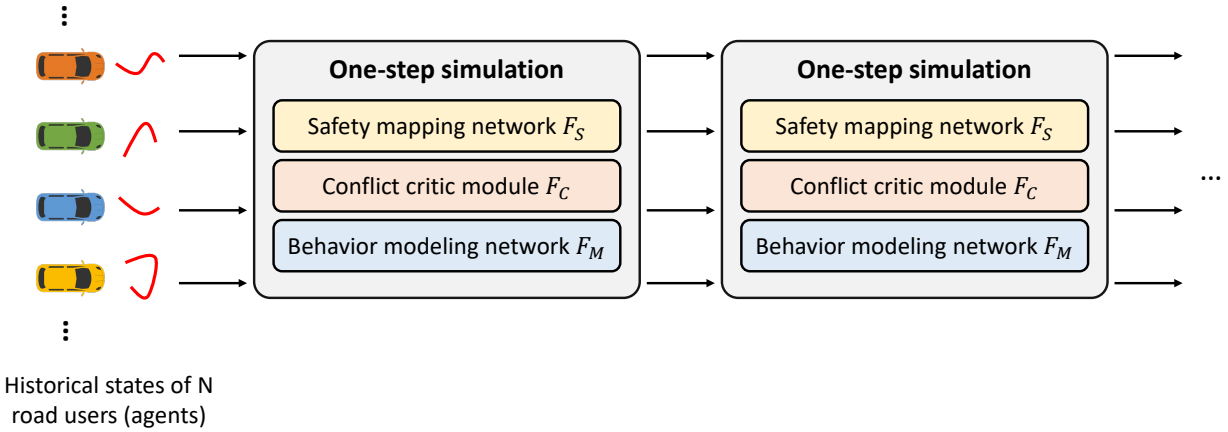


Figure 3.3: Illustration of the simulation process

The differentiable safety mapping network is a neural mapper pretrained from physics and driving rules to map unsafe behaviors to a feasible domain of safety. Therefore, the safety mapping

network will guide vehicle behavior and rectify their actions in safety-critical situations. The safety mapping network is pretrained and fixed when training the behavior modeling network and the discriminator jointly. During the simulation process, as shown in Figure 3.3, the state of all road users will be updated based on the behavior modeling network, conflict critic module, and the safety mapping network in each simulation step to autoregressively generate the simulation environment. The details of each component in the NeuralNDE framework will be introduced in the following sections.

3.2.2 Behavior modeling network

We frame the behavior modeling via IL with the help of large-scale real-world offline demonstrations. Given a large-scale collection of real-world vehicle trajectory data, we aim to jointly model both vehicle-to-vehicle interactions and their long-term state trajectories within a certain temporal range. In our framework, we consider each vehicle instance as an agent with stochastic actions and future states where the actions and states of each agent are not only related to its own historical trajectories, but also to that of all other agents.

Suppose s_i^t and a_i^t represent the state vector (e.g., location, pose, vehicle size, etc.) and action vector (e.g., acceleration, yaw rate, etc.) of ith agent at time step t. $S_N^{\tau:t} = \{s_1^{t-\tau+1}, \dots s_1^t, \dots s_N^{t-\tau+1}, \dots s_N^t\}$ represents a collection of the state trajectories of all N agents from all τ time steps ahead of the current time t. The modeling of all agents' future actions can be thus essentially considered as a conditional probabilistic inference problem, i.e., to estimate the joint distribution of actions from all agents $p(a_1^t, \dots, a_N^t | S_N^{\tau:t})$ given their historical states as conditional inputs. To accurately model the joint distribution, the Transformer model is used as the backbone of our behavior modeling network. Transformer models originated from the field of natural language processing (Vaswani et al., 2017), and have revealed remarkable performance in many applications, including computer vision (Dosovitskiy et al., 2020), bioinformatics (Jumper et al., 2021), and multimodal data generative modeling (Ramesh et al., 2021).

There are three advantages to modeling each agent as a "token" in the language model. The first advantage is that the Transformer is naturally suitable for modeling long-term interactive behavior in a multi-agent environment. The self-attention mechanism is capable of characterizing inter-token relations, which model the interaction between agents. The position-wise feed-forward network in the Transformer can capture intra-token information, which measures the influence of the historical states of each agent on their future behavior. The second advantage is model scalability. The Transformer can easily handle a large number of tokens (e.g., hundreds to thousands), which allows our framework to scale to large road networks with a large number of road users. The third advantage is the permutation invariant property. The Transformer block is

permutation invariant to the order of tokens. Therefore, by modeling each agent as a token, we do not need to specify the order of agents, which are geographically located in a two-dimensional space (i.e., on a road), making it difficult to order them in a one-dimensional space (i.e., determine the token order in input).

At each time step of modeling, the behavior modeling network F_M takes in the historical states $S_N^{\tau:t}$ of all agents and is trained to jointly predict their future actions $(a_1^t, \ldots, a_N^t)$. Instead of predicting deterministic actions values, we predict the stepwise action distributions and consider distribution as a multi-variable Gaussian over their action space:

$$p\left(S_N^{\tau:t}\right) = F_M\left(S_N^{\tau:t}\right) \sim N\left(\mu_{a,i=1\ldots N}^t, \Sigma_{a,i=1\ldots N}^t\right), \tag{3.1}$$

where $p\left(S_N^{\tau:t}\right)$ is the joint action distribution and $\mu_{a,i=1\ldots N}^t$ and $\Sigma_{a,i=1\ldots N}^t$ are the mean and covariance matrix of the Gaussian distribution. After we obtain the joint distribution of actions, a group of action vectors for each agent are sampled:

$$a_1^t, \ldots, a_N^t \leftarrow N\left(\mu_{a,i=1\ldots N}^t, \Sigma_{a,i=1\ldots N}^t\right). \tag{3.2}$$

Then, for each agent, its new state vector s_i^{t+1} is determined by a differentiable state transition function T determined by vehicle dynamics:

$$s_i^{t+1} = T\left(a_i^t, s_i^t\right). \tag{3.3}$$

The above processing will be repeated so that new states of all agents can be generated in an autoregressive manner. In practice, instead of one-step prediction, multiple time steps (e.g., κ steps) predictions $S_N^{t:\kappa} = \{s_1^{t+1}, \ldots s_1^{t+\kappa}, \ldots s_N^{t+1}, \ldots s_N^{t+\kappa}\}$ will be made by the behavior modeling network. Note that to simulate the uncertainty of drivers, during the simulation, at each time step, we will sample from the joint distribution to determine all vehicles' future trajectories and then simulate forward. Also, our model can be easily extended to generate multimodal outputs, where several Gaussian distributions instead of one will be predicted to further improve the uncertainty of drivers (Chai et al., 2019).

Human driving behavior is influenced by traffic signals. Therefore, to incorporate traffic signal information into the model, we can treat each traffic signal head as a dummy agent, with its state represented in a similar format, encompassing information like position and traffic light status (e.g., green, red, yellow, etc.). This unified approach allows us to model various agents consistently, where tokens can represent either road users (e.g., vehicles) or traffic signal heads. For the sake of a more concise and clear mathematical derivation, we will omit the traffic signal information from the input vector. However, please note that these inputs will be integrated into the input state

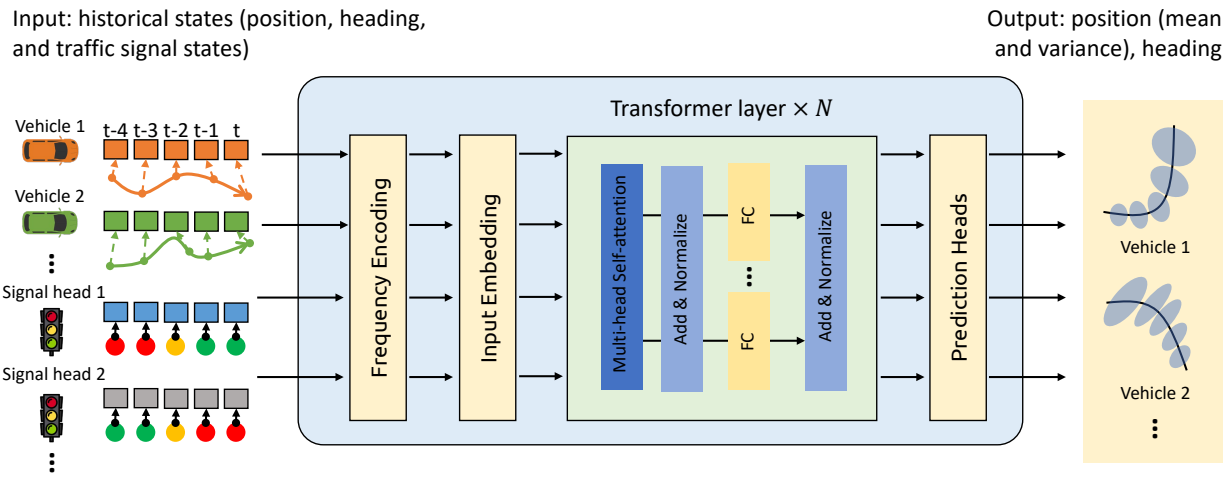


Figure 3.4: Network architecture of the behavior modeling network

vector when modeling signalized intersections.

The proposed behavior modeling network consists of a frequency encoding layer, an input embedding layer, a Transformer backbone, and a prediction layer, as shown in Figure 3.4. The input to the behavior modeling network is all tokens (including both road user and traffic signal head) historical states, and the output is the joint state distribution of road users. Detailed network architecture is discussed in the next section. The input embedding layer is a fully connected layer with weights shared across different tokens. The Transformer backbone consists of several standard BERT (Devlin et al., 2018) layers stacked on top of each other. Since the action prediction is independent of the input order of the N agent, we, therefore, have removed the "positional encoding", which is a standard encoding layer in Transformers to capture order-related information for sequence input data. Also, before the state vectors are input to the input embedding layer, we adopt the ideas of follow Mildenhall et al. (2021) using to use frequency encoding, which applies a set of sine and cosine basis functions that projects the vectors to high dimensional space to improve capturing high-frequency variation in the state spaces. Suppose γ defines a mapping function from R^1 to R^{2L+1}, where L is the order of frequencies. The state value s after mapping can be written as follows:

$$\gamma(s) = \left[s, sin\left(2^0\pi s\right), cos\left(2^0\pi s\right), \ldots, sin\left(2^{2L-1}\pi s\right), cos\left(2^{2L-1}\pi s\right)\right]. \tag{3.4}$$

To train the model F_M with implicit variance, in the prediction layer, two prediction heads are attached for each input token at the output end, one for predicting $\mu^t_{a,i}$, another for predicting $\Sigma^t_{a,i}$. The training of the behavior modeling network can be formulated as a maximum likelihood estimation process. Given N agents of T time steps, the state trajectories within $[t-\tau, t]$ are used

as input and the action vectors at time t are used as the ground truth ($t = 1, \ldots, T$), then the likelihood function can be written as follows:

$$p\left(F_M\right) = p\left(s_1^1, s_1^2, \ldots, s_i^t, \ldots, s_N^T\right). \tag{3.5}$$

For simplification, we assume that there is no correlation between variables in the multivariate Gaussian distribution, so the action covariance matrix for each agent is a diagonal matrix, i.e., $\Sigma_{a,i}^t \approx diag\left(\sigma_{i,1}^t, \ldots, \sigma_{i,D}^t\right)$, D is the dimension of the action vector. Then the joint probability of an action vector a_i^t can be implied as follows:

$$p\left(s_i^t\right) = \prod_{j=1}^{D} \frac{1}{(2\pi)^{\frac{1}{2}}} \frac{1}{\sigma_{i,j}^t} exp\left\{-\frac{\left(\mu_{i,j}^t - a_{i,j}^t\right)^2}{2\left(\sigma_{i,j}^t\right)^2}\right\}, \tag{3.6}$$

where $\mu_{i,j}^t$ represents the predicted jth action value at time t for ith agent.

We approximate the joint probability distribution $p\left(F_M\right)$ in Equation 3.5 as the multiplicative form of each agent's marginal probabilities, and combine it with Equation 3.6 to derive the loss function in the negative log-likelihood form as follows:

$$L_M\left(F_M\right) = \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{D} \left[\ln\left(\sigma_{i,j}^t\right) + \frac{\left(\mu_{i,j}^t - a_{i,j}^t\right)^2}{2\left(\sigma_{i,j}^t\right)^2}\right]. \tag{3.7}$$

The approximation of Equation 3.5 will not affect the solution since the term $\frac{1}{2}\left(\mu_{i,j}^t - a_{i,j}^t\right)^2 / \left(\sigma_{i,j}^t\right)^2$ in Equation 3.7, which represents the expected prediction accuracy of the actions, can still make the model converge to the optimal solution. Note that although we don't have ground truth for the predicted variance $\sigma_{i,j}^t$, it can be jointly estimated as implicit variables along with the mean action $\mu_{i,j}^t$ during the training process, where a high uncertainty prediction naturally responds to a large variance and vice versa.

3.2.3 Conflict critic module

The simulated environment must be able to reproduce accurate safety-critical driving statistics including both near-miss and crash events. Although the behavior modeling network can generate realistic conflicts, it may not be able to achieve distribution-level accuracy, due to the rarity of safety-critical events in the training dataset. For example, the crash rate can be unrealistically high, and the crash type distribution can be inconsistent with the real-world driving environment. To tackle this issue, we design a model-based conflict critic module F_c to control the occurring frequencies and patterns of safety-critical behaviors during the inference time to achieve statistical realism, as illustrated in Figure 3.2. The input to F_c are the sampled κ steps predicted trajectories of

all N agents $S_N^{t:\kappa} = \{s_1^{t+1}, \dots s_1^{t+\kappa}, \dots s_N^{t+1}, \dots s_N^{t+\kappa}\}$ generated by the behavior modeling network at the current time t. The output of F_c is the acceptance probability p_a for not passing through the safety mapping network:

$$p_a = F_c\left(S_N^{t:\kappa}\right). \tag{3.8}$$

If there is a potential conflict in predicted trajectories $S_N^{t:\kappa}$, we will have a probability p_a to accept it, and a probability $1 - p_a$ to reject it and let the safety mapping network guide and rectify the dangerous driving behavior. The acceptance probability is trajectory-dependent, which means that for those conflict patterns that have a higher probability of occurring in the real world, we will have correspondingly higher p_a to accept it. Therefore, by calibrating the F_c function, we can control the generation process of safety-critical events to match real-world statistics in both near-misses and crashes. Specifically, in the chapter, each crash type will have a specific acceptance probability. The implementation details and calibration methods are introduced in the following paragraphs.

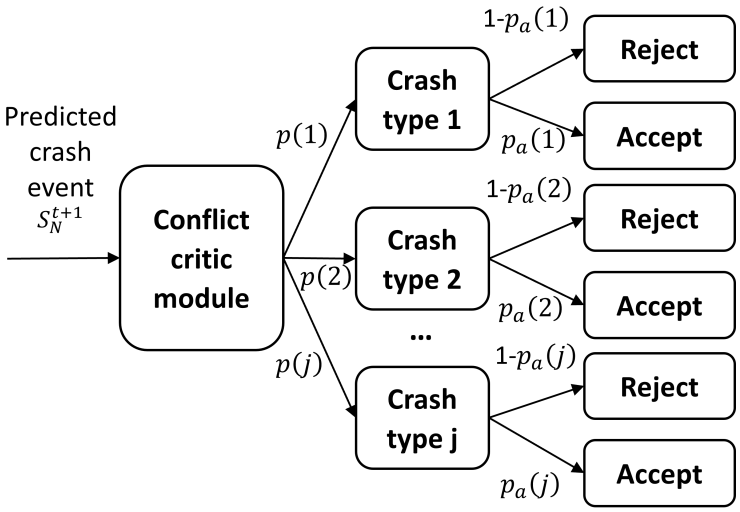


Figure 3.5: Illustration of the conflict critic module

In this chapter, we consider vehicle conflicts in a one-step prediction for simplicity. Let S_N^{t+1} denote all vehicle states predicted by the behavior modeling network at the next time step $t + 1$. If there is a crash happening in the predicted trajectory, we will have a certain probability to accept the crash and generate it, otherwise, the vehicle behavior will be rectified by the safety mapping network to avoid the crash. The acceptance probability will depend on the predicted crash type that happens in S_N^{t+1} and will be calibrated as discussed in the next paragraph. For the same crash type, the acceptance probability will be the same. If there is no crash in S_N^{t+1}, the acceptance probability will be zero. An illustration figure is shown in Figure 3.5. By calibrating the conflict critic module,

i.e., obtaining the acceptance probability $p_a(j)$ for different crash types j, we can realize accurate crash rate and crash type distribution of the simulation environment.

The calibration process is divided into two steps, where the first step aims to fit the crash rate and the second step tries to fit the crash type distribution. In the first step, we first assume a uniform acceptance probability (p_{ua}) for different crash types and try to fit the ground-truth crash rate. The calibration process is, at the 1st iteration, making a random initial guess of the uniform acceptance probability $p_{ua}^1 \in (0, 1]$, then run simulations to obtain the current NeuralNDE crash rate at the 1st iteration c^1. Then linearly update the uniform acceptance probability as follows

$$p_{ua}^{i+1} = c^{gt} \cdot \frac{p_{ua}^i}{c^i}, \tag{3.9}$$

where c^{gt} denotes the desired ground-truth crash rate, and i denotes the current iteration number. Continue this process until the NeuralNDE crash rate is close to the ground truth with satisfactory accuracy. In the second step, we will calibrate the acceptance probability for each crash type. The acceptance probability $p_a(j)$ for crash type j needs to satisfy the following system of linear equations to fit both crash rate (Equation 3.10) and crash type distribution (Equation 3.11):

$$\sum_j p(j)\, p_a(j) = p_{ua}. \tag{3.10}$$

$$\frac{p(j)\, p_a(j)}{\sum_j p(j)\, p_a(j)} = c^{gt}(j), \forall j \in J, \tag{3.11}$$

where $c^{gt}(j)$ is the ground-truth probability of crash type j, p_{ua} is the uniform acceptance probability obtained from the first step, and $p(j)$ is the probability of crash type j occurring in NeuralNDE using the uniform probability p_{ua}. For Equation 3.10, the summation of $p(j)\, p_a(j)$ over all potential crash types j denote the overall acceptance probability considering different crash types. It needs to be equal to the uniform acceptance probability (p_{ua}) obtained in the first step, which can guarantee the accurate crash rate of the simulation. Therefore, the acceptance probability $p_a(j)$ equals to

$$p_a(j) = p_{ua} \cdot \frac{c^{gt}(j)}{p(j)}. \tag{3.12}$$

We will use $p_a(j)$ as the acceptance probability of different crash types $j \in J$ for the conflict critic module.

3.2.4 Safety mapping network

To improve the modeling accuracy and achieve statistical realism in safety-critical conditions, we propose a safety mapping network that can guide vehicle behavior in safety-critical situations by mapping the unsafe vehicle behaviors to their closest safe neighbors. The safety mapping network serves as a safety guard to rectify vehicle behaviors before an imminent crash. Given the current state and predicted κ steps future actions of all agents $\{S_N^t, A_N^{t:\kappa}\}$, the safety mapping network F_S jointly predicts the rectified actions $A_N^{t:\kappa,\star}$ of all agents as follows

$$A_N^{t:\kappa,\star} = F_S\left(S_N^t, A_N^{t:\kappa}\right). \tag{3.13}$$

If there is an impending crash using the original action vector $A_N^{t:\kappa}$, the safety mapping network will modify the action vector to resolve the potential conflict. Note that the action rectification will only be done if the original action vector will result in a predicted crash, otherwise the action output by the safety mapping network will be the same as the original action vector.

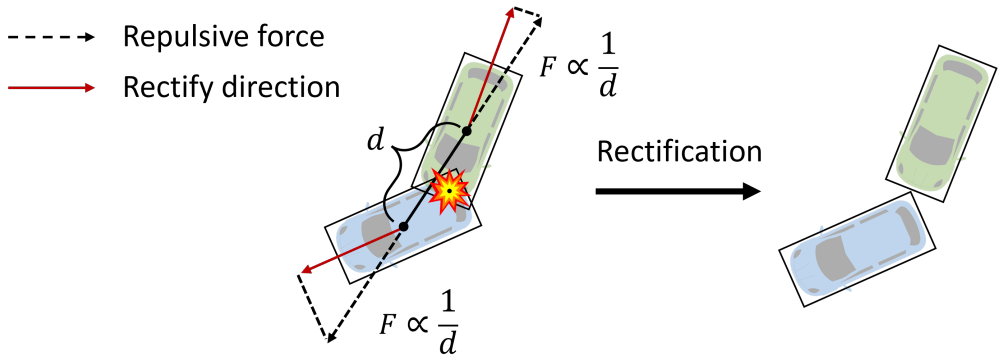


Figure 3.6: Illustration figure of the physics-based safety mapping rule to guide vehicle behavior in safety-critical situations

The safety mapping network is trained to imitate existing model-based safety guards based on domain knowledge. In this chapter, for simplicity and generality, we consider one-step prediction and use a physics-based safety guard as the training target. The illustration figure is shown in Figure 3.6. When two vehicles are going to collide with each other, we resolve the potential conflict by setting a repulsive force between them. The force is projected to the heading direction of each vehicle and restricts their action to avoid the crash. We generate a large number of offline random states-response pairs based on the above rules. The loss function for training the safety mapping network can be written as follows:

$$L_S\left(F_S\right) = \left\| F_S\left(S_N^t, A_N^t\right) - \hat{A}_N^t \right\|_1. \tag{3.14}$$

where $\hat{A}_N^t$ are the ground truth rectified actions of each agent at time t, $\|\cdot\|_1$ represents the sum of element-wise absolute distance. Since the action rectification also involves complex interactions between agents, we also use the Transformer as the backbone of the safety mapping network. Similar to the behavior modeling network, each agent is considered as an individual token, and the Transformer is trained to predict the residue between the rectified and the reference control. After training, the pretrained safety mapping network will be fixed and embedded into the framework, therefore, the whole pipeline can be trained end-to-end as shown in Figure 3.1.

By incorporating the safety mapping network, we can mitigate the inevitable modeling error of the behavior modeling network in safety-critical situations. We showed that the safety mapper significantly reduces the modeling error (e.g., measured by crash rate) by several orders of magnitude in Ablation studies, while such behavior is extremely difficult for existing data-driven approaches to master due to the "curse of rarity" issue discussed previously. Also, it helps to decouple the safety objective when training the behavior modeling network and let it focus on realistic multi-agent interaction modeling. It should be noted that the proposed method is not limited to the chosen physics-based rule. Different safety guards proposed recently can also be used, for example, safety envelope-based methods (Shalev-Shwartz et al., 2017), potential force field-based methods (Wang et al., 2015; Nistér et al., 2019), online verification methods (Pek et al., 2020), etc.

3.2.5 Generative adversarial training

To further improve the realism of the generated trajectories and tackle the distribution shift issue, generative adversarial training is adopted when training the behavior modeling network. The key to the generative adversarial training is a minimax two-player game under which two networks will contest with each other and force the generated data to be indistinguishable from real ones (Goodfellow et al., 2020). During the training, we rollout forward the simulation for several steps and assume the generated trajectories can be easily differentiated from real ones if they exhibit unrealistic patterns (e.g., offroad or other distribution shift behaviors). To this end, we introduce a discriminator network – a Multilayer Perceptron (MLP) network, which takes in the state trajectories of an agent and is trained to distinguish whether the input is sampled from the real-world dataset or from the simulation. Meanwhile, we force the behavior modeling network to capture the true distribution of real trajectories and make generated data indistinguishable from the discriminator side. In this way, the adversarial loss can be backpropagated to the behavior modeling network to further improve the modeling fidelity.

Suppose D represents the discriminator network, $\hat{S} \sim p_R\left(S_N^t\right)$ represents a trajectory sampled from the real-world data distribution, and $S \sim p_G\left(S_N^t\right)$ represents a trajectory generated from

the simulation. We follow a standard adversarial training pipeline and define the adversarial loss functions as follows:

$$L_{adv}\left(F_M, D\right) = E_{\hat{S}\sim p_R\left(S_N^t\right)}\left[\log D\left(\hat{S}\right)\right] + E_{S\sim p_G\left(S_N^t\right)}\left[\log\left(1 - D\left(S\right)\right)\right]. \tag{3.15}$$

During the training process, since all components are differentiable, the networks F_M and D can be alternatively updated under a unified objective. By combining the loss function (Equation 3.7) of the behavior modeling network F_M, our final objective function is defined as follows:

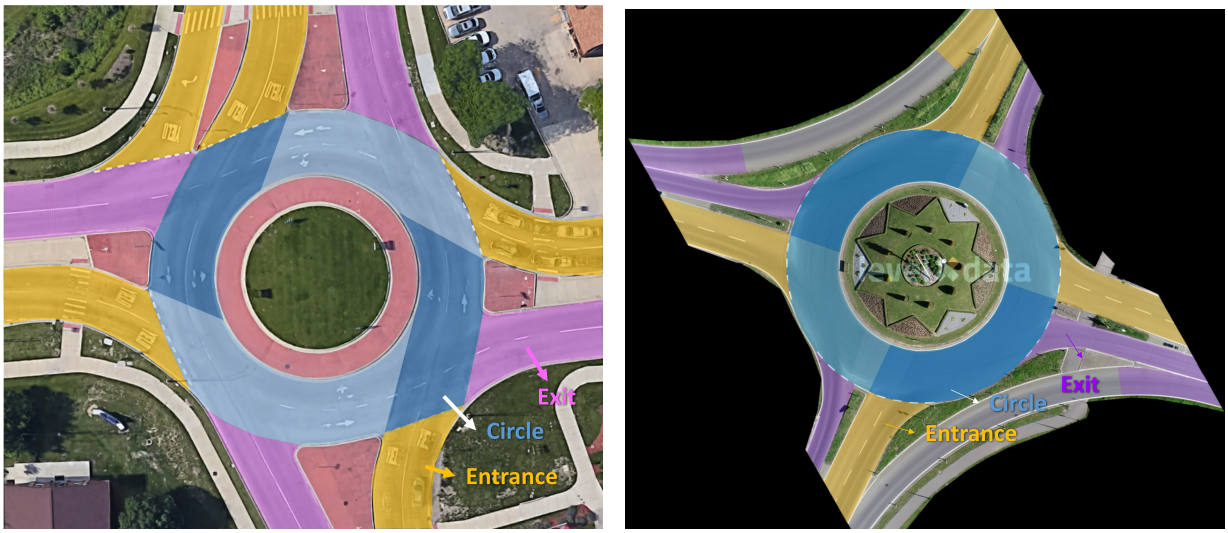
$$F_M^\star, D^\star = \arg\min_{F_M}\max_{D}\left[L_M\left(F_M\right) + \beta L_{adv}\left(F_M, D\right)\right]. \tag{3.16}$$

where F_M tries to minimize this objective while D tries to maximize it. β is a pre-defined hyperparameter for balancing the weights between the two loss terms.

3.3 Implementation details

3.3.1 Datasets

Roundabout is an important and challenging urban driving environment for AVs. We validate our model using a real-world dataset collected from a two-lane roundabout located at State St. and W Ellsworth Rd. intersection, Ann Arbor, Michigan, USA (abbreviated as AA dataset). The illustration figure of this two-lane roundabout is shown in Figure 3.7a. This is a busy roundabout with a large traffic volume and the fourth highest crash rate in Michigan (Gursten, 2021). A roadside perception system (Zhang et al., 2022b; Zou et al., 2022) is deployed for real-time traffic object detection, localization, and tracking to collect all vehicle trajectory information (e.g., position, heading) within the roundabout at 2.5Hz. The AA dataset includes both the detailed normal and safety-critical driving conditions data. The safety-critical events data, which includes crash event trajectories, crash videos, police crash reports, etc., are crucial for providing safety-critical statistics ground-truth to validate the simulation fidelity. To the best of our knowledge, the real-world safety-critical rare-event data are not available in any most existing public datasets, however, they are essential for constructing and validating the performance of generated simulation environments. For training purposes, we used data collected on May 2nd, 2021, from 10:00 to 17:00, including around 17,000 road users. For each vehicle, the data includes its position, heading, and other information at 2.5 Hz. We excluded frames that involve pedestrians, cyclists, and trailers since there are only a few frames that include these agents and the data size is limited for training. It should be noted that the proposed method can handle diverse road users (e.g., pedestrians) and model their interactions if the data is sufficient. For validation purposes, we



(a) (b)

Figure 3.7: Illustration figure of the studied location: (a) Roundabout at Ann Arbor, Michigan, USA. (b) Roundabout at Neuweiler, Aachen, Germany

used crash data from large-scale trajectories and police crash reports (Michigan Office of Highway Safety Planning, 2022) to obtain ground-truth safety-critical events statistics (e.g., crash rate and crash type distribution).

We also use an open dataset, rounD (Krajewski et al., 2020) to further demonstrate NeuralNDE performance in normal driving conditions. The illustration figure of this two-lane roundabout is shown in Figure 3.7b. The rounD dataset is collected at three different locations with high accuracy tracking of around 13,000 road users at high frequency (25Hz). We chose the roundabout with most of the data and it is located at Neuweiler, Aachen, Germany. Similar to the AA dataset, we also exclude frames that involve pedestrians, cyclists, and trailers for training. We will use results on AA datasets to illustrate the performance and the results on rounD dataset are similar and can be found in Appendix B.1.

3.3.2 Network architecture

The network architecture of the behavior modeling network is shown in Figure 3.4. Each road user and traffic signal head is considered a token and the input to the network is the historical state, i.e., position (x, y coordinates), heading (cosine and sine of heading), and traffic light state (one-hot encoding for green, red, yellow, and unknown), of all tokens within the historical time window. If a token is a road user (e.g., vehicle) then the traffic light state info is padded with zero, and vice versa. The number of historical steps at the input is set to $\tau = 5$ with a resolution of 0.4 seconds per step. Then, the input will pass through the frequency encoding layer (Mildenhall et al.,

2021), in which we apply a set of sine and cosine basis functions that projects the vectors to high dimensional space to improve capturing high-frequency variation in the state spaces. Suppose γ defines a mapping function from R^1 to R^{2L+1}, where L is the order of frequencies, we use $L = 4$ in the chapter. The input embedding layer is a fully connected layer that converts the state dimension to the Transformer hidden layer dimension. Then a set of ($N = 4$) standard BERT (Devlin et al., 2018) transformer layers is stacked together. The dimension of the hidden layer in the Transformer block is 256, the number of heads in multi-headed attention layers is 4, the dimension of intermediate layers in the position-wise feedforward net is 512, the probability of dropout of various hidden layers is 0, and the probability of dropout of attention layers is 0. Then, the output from the Transformer will pass through prediction heads (single layer MLP with 256 neurons) to generate the final output. In practice, we directly predict the states of each vehicle rather than actions for simplicity. Therefore, the output of the behavior modeling network is the predicted trajectory, i.e., stochastic position (one prediction head for the mean of position, one prediction head for the variance of position, and one prediction head for deterministic heading), of each vehicle in the prediction time horizon. The number of prediction steps at the output is set to $\kappa = 5$ with a resolution of 0.4 seconds per step.

The discriminator is a four-layer MLP with dimensions $1024 \star 512 \star 256 \star 1$. The activation function is LeakyReLU with a slope equal to 0.2. The input of the discriminator is the trajectory either from the behavior modeling network or the real-world sample and the output is a scalar value. Similar to the behavior modeling network, frequency encoding is applied before passing through the MLP.

The network architecture of the safety mapping network is the same as the behavior modeling network. The input of the network is also the position and heading of all vehicles. The safety mapping network is performed frame-by-frame, so the input includes only the vehicle state at the current step. The output of the safety mapping network is the position and heading after rectifications that project the unsafe state to the nearest safe one.

3.3.3 Training details

We train the safety mapping network by using the RMSprop (Hinton et al., 2012) optimizer. We set the batch size to 64 and the learning rate to 0.0001. The learning rate is reduced to its 0.3 every 600 epochs. The training took around 20 days on an Intel i7-10700F CPU and NVIDIA 3070 GPU desktop with a total number of 3,000 training epochs. To cover all potential safety critical patterns, we randomly sampled the vehicle states as input and their ground truth is generated with a rule-based model. When two vehicles are going to collide, we push them apart by setting a repulsive force between them. The force is projected to the heading direction of each vehicle and rectifies

their states until they are not colliding with each other, as illustrated in Figure 3.6. Each vehicle is considered as 3.8 meters in length and 2.0 meters in width when training the safety mapper, which includes a 0.2 meters buffer compared to the real size. Note that we do not modify the heading of each vehicle and only rectify the position, which is similar to guiding the vehicle to decelerate or accelerate in safety-critical situations to avoid a crash. Instead of directly predicting the rectified states, we train the mapper to generate the residual between the ground truth and the input. The rectification is performed frame-by-frame. The mean absolute error between the predicted position residue and the ground-truth residue is used as the loss function. Since safety-critical situations rarely happen, the residual may follow a sparse pattern where most of the values are close to zero. Therefore, when generating the training data, we balance the ratio between the activated and non-activated output by using heuristic sampling where in each frame, the first 80% of vehicle states are uniformly sampled and the rest 20% are sampled from the neighbor of existing vehicles. We generate 240,000 random frames for each training epoch. During the training phase, the number of tokens (vehicles) is set to a fixed number of 32 considering the batch-wise training efficiency. However, in the inference phase, there are no such restrictions, and the network can adapt to any number of vehicles.

When training the behavior modeling network, we freeze the safety mapping network. Both the behavior modeling network and the discriminator are updated jointly by using the RMSprop (Hinton et al., 2012) optimizer. The batch size is set to 32 and the learning rate is set to 0.0001 with decay to its 0.3 every 300 epochs. We set the training token size to 32. When there are fewer than 32 vehicles in the road network, fake vehicle states will be used to pad the input matrix. Data augmentation is applied with Gaussian noise of zero mean and 0.0025 variance for position and 0.000001 for cosine and sine of heading. The number of training epochs is set to 1,500 and the training takes around 3 days on an NVIDIA 3070 GPU desktop. The number of historical steps at the input is set to 5 with a resolution of 0.4s per step. The number of output steps within a single forward pass is set to 5 with the same resolution. In practice, we train the network to predict the states rather than the actions. The state variables include the position (x and y coordinates) and heading (cosine and sine heading) of each vehicle. The loss function is composed of three parts: imitation loss of position, imitation loss of heading, and adversarial loss. The weight of each component is set as 1, 20, and 0.1. The imitation loss of position and heading is calculated by the mean absolute error between the predicted states (predicted x and y coordinates and heading) and ground-truth states at the next 5 steps. The adversarial loss is calculated using the BCEWithLogitsLoss following the general setting of generative adversarial training.

3.4 Case studies

3.4.1 Experiment settings

The proposed NeuralNDE simulator is first initialized with a randomly sampled trajectory clip of 2 seconds with all agents following their logged trajectories. Then, all agents' behaviors are controlled by NeuralNDE. At each simulation time step, new vehicles will be generated in each entry lane by following a Poisson process whose arrival rate is calibrated using the dataset. Also, vehicles will leave the road network when reaching exit areas. We assume all vehicles have an identical size with 3.6 meters in length and 1.8 meters in width. Note that the proposed method can be easily extended to handle different vehicle sizes by incorporating length and width in the input data. Each simulation episode lasts for 3600 seconds with a simulation resolution of 0.4 seconds. If a crash happens, the simulation will be terminated early. We use around 15,000 simulation hours of data to validate the statistical realism of the NeuralNDE, where all data are used for calculating crash-related metrics and 100 hours of data are used for other metrics. We conducted the experiments on the University of Michigan's Great Lakes High-Performance Computing (HPC) cluster using 1000 cores and 2000 GB RAM. It took around 1440 seconds of real-world time to conduct 3600 seconds of simulation. Therefore, the simulation speed ratio (simulation time/real-world time) is around 0.4.

We compare the proposed method with SUMO (Lopez et al., 2018) - a widely used simulation platform for traffic environments, and other state-of-the-art methods. For SUMO simulator, the map is obtained from the OpenStreetMap OpenStreetMap contributors (2017). For each episode, the simulation duration and time resolution are the same with NeuralNDE. The Sublane-Model is used to improve the simulation fidelity since by default vehicle lane changes are performed instantly and vehicles are always staying on the centerline of the road in SUMO. The lateral resolution is set as 0.25m for the continuous lane-change behavior. The IDM Treiber et al. (2000) and SL2015 model are used as the car-following model and lane-changing model, respectively. More details for the SUMO simulator settings can be found in Appendix B.2.

3.4.2 Evaluation metrics

To evaluate the fidelity and statistical realism of the proposed NeuralNDE, a suite of statistical metrics is examined, with both normal and safety-critical driving behaviors. The metrics include:

1. Vehicle instantaneous speed distribution;
2. Vehicle distance distribution;
3. Vehicle yielding distance and yielding speed distributions;

4. Traffic volume distribution;
5. Vehicle origin-destination (OD) distribution;
6. Vehicle crash rate;
7. Vehicle crash type distribution;
8. Vehicle crash severity distribution;
9. Vehicle Post-Encroachment Time (PET) distribution.

The instantaneous speed distribution is collected when vehicles travel in the roundabout circle. The speed is calculated using the Euclidean distance traveled between two timesteps divided by the simulation time resolution. To measure the distance between two vehicles, each vehicle is approximated using three circles with an equal radius as shown in Figure 3.8. Vehicle distance is defined by the nearest circle centers of two vehicles. We use $r = 1.0$ meters and $l = 2.7$ meters in this chapter.

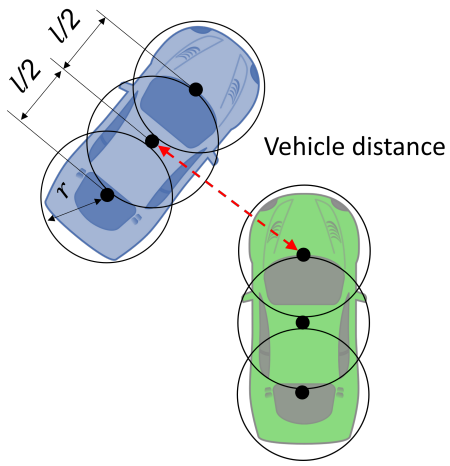


Figure 3.8: Illustration figure of the vehicle distance

A vehicle is considered to yield if it reaches a running stop, i.e., speed smaller than 5mph, in the yielding area of each entry as shown in Figure 3.9. Vehicles in the corresponding circle quadrant as shown in Figure 3.9 are conflicting vehicles for the vehicle in the yielding area. The vehicle yielding distance is the Euclidean distance between 1) the yielding vehicle at the entrance and 2) the nearest conflicting vehicle in the roundabout. The speed of the closest conflicting vehicle is recorded for the vehicle yielding speed distribution. The traffic volume and vehicle origin-destination are obtained from vehicle trajectories.

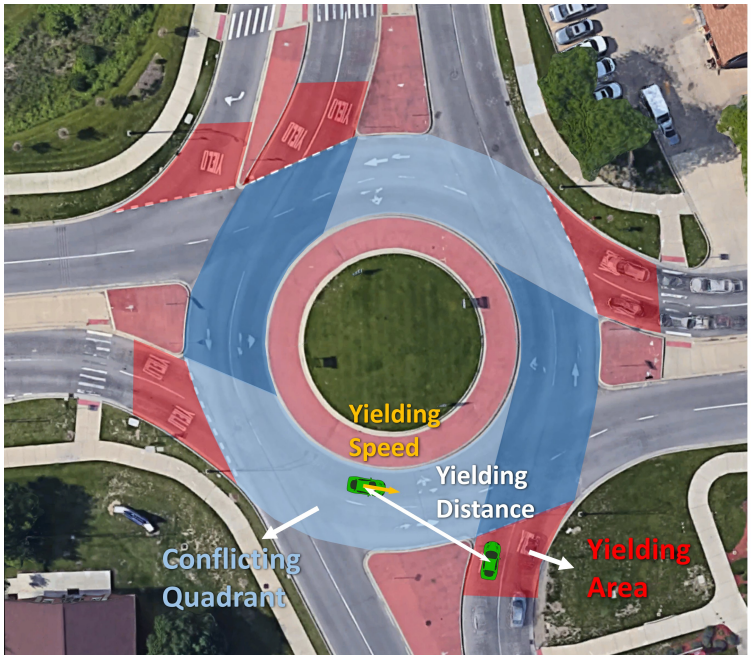


Figure 3.9: Illustration figure of the yielding area

Two agents are considered in a crash if their bounding boxes overlap. The crash rate is calculated by the number of collisions divided by the total travel distances of all vehicles. The crash type is adopted from the definition of the National Highway Traffic Safety Administration (2022).

We use the Change in Velocity (Delta-V), a widely used metric to estimate occupant injury risk to measure the simulated crash severity. It is defined by the difference between the vehicle impact speed and the separation speed. The impact speed is the vehicle speed at the crash moment, and the separation speed is calculated based on the conservation of momentum. Then based on the Delta-V we can obtain the occupant injury level. More details of the determination of crash type and crash severity can be found in Appendix B.3.

The PET is a widely used surrogate safety measure for characterizing near-miss events. It is defined by the time difference between a vehicle leaving the potential conflict area and a conflicting vehicle entering the same area. We will only consider the PET within the roundabout circle where most conflicts happen. We rasterize the roundabout into 1.3×1.3 meters blocks, and each block is a potential conflict area.

We compare the statistics between the simulated results and the empirical ground truth data. To quantitatively measure the divergence between two distributions, Hellinger distance and Kullback–Leibler Divergence (KL-divergence) are used as measurements. For two discrete

probability distributions P and Q, their Hellinger distance D_H is calculated as follows:

$$D_H(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_x \left(\sqrt{P(x)} - \sqrt{Q(x)}\right)^2}, \tag{3.17}$$

which is directly related to the Euclidean norm of the difference between the square root of the two probability vectors. The range of Hellinger distance is between 0 to 1, and the smaller the value, the more similar the two distributions. Suppose P is the real-world data distribution and Q is the simulated distribution, the KL-divergence D_{KL} can be calculated as

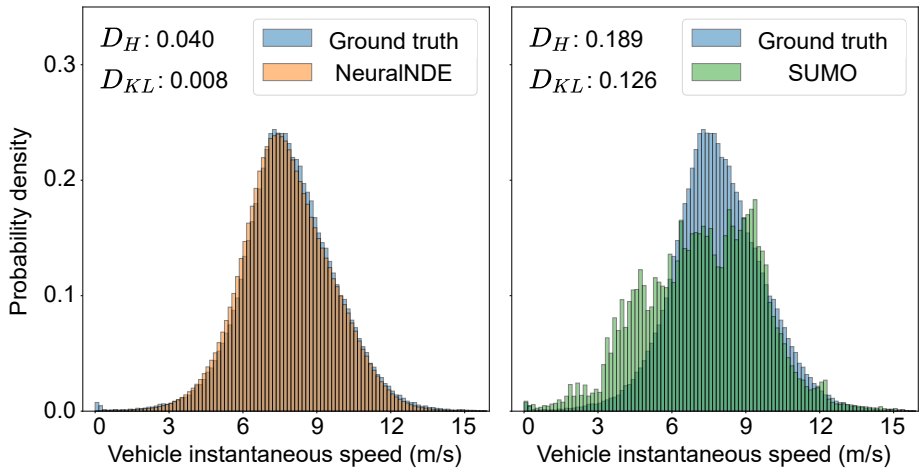
$$D_{KL}(P,Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}. \tag{3.18}$$

KL-divergence ranges from 0 to infinity, and also the smaller the value, the more similar the two distributions.

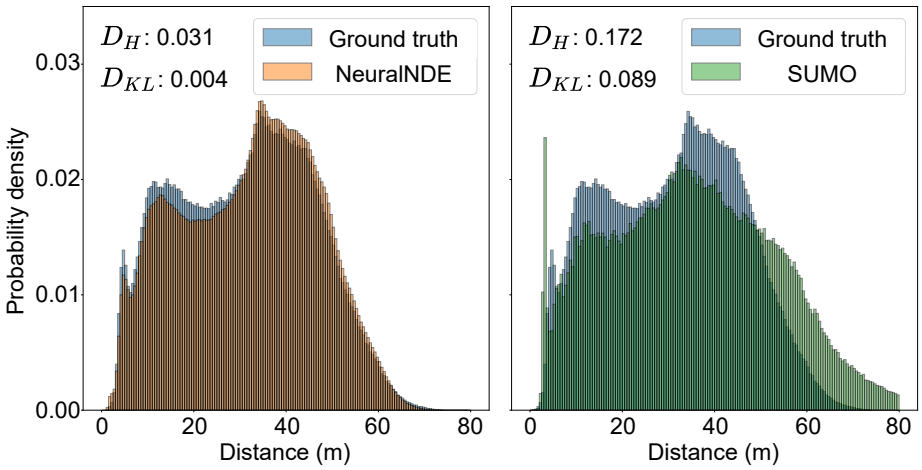
3.4.3 Statistical realism of normal driving behavior

Since high-fidelity normal driving behavior is the prerequisite for reproducing accurate safety-critical events, in this section, we will first validate the statistical realism of normal driving statistics of the proposed NeuralNDE. Vehicle speed and position are direct outcomes of microscopic driving behaviors, and they are critical for both training and testing the AV. The proposed NeuralNDE can generate accurate vehicle instantaneous speed distribution as in the real world, as shown in Figure 3.10a. Compared with the SUMO baseline, vehicle speeds in NeuralNDE are naturally distributed among the whole range, covering both low and high-speed situations. Furthermore, NeuralNDE can also accurately reproduce vehicle distance distribution as shown in Figure 3.10b, reflecting the full distribution of encounters that AV might face in the real world.

Traffic volume and the origin-destination of each vehicle are also important statistics that reflect the overall traffic condition of the environment. The traffic volume distribution of the simulated environment is consistent with the real world as shown in Figure 3.11a. The vehicle origin-destination (OD) distribution of each entry is shown in Figure 3.11b. The simulated vehicles exhibit a similar travel pattern as in the real world. Note that the destination of each simulated vehicle is not predetermined manually, where each vehicle will imitate real-world driving behaviors and choose its desired exit by progressively sampling from the distribution of future positions. For example, consider a vehicle just entering the roundabout from the west entrance, since there are only a small amount of real-world trajectories that turn right and take the first exit, this vehicle will have a high probability to continue driving in the roundabout and choose to take other exits following the real-world distribution. Therefore, we can find that the



(a)



(b)

Figure 3.10: Statistical realism of normal driving behavior. (a) Vehicle instantaneous speed distribution. (b) Vehicle distance distribution.

proposed NeuralNDE can replicate accurate traffic flow characteristics.

In a two-lane roundabout environment, a highly interactive location is at the roundabout entrance where entering vehicles need to yield to conflicting vehicles within the roundabout. Many real-world conflicts and crashes occur in this location, and the fidelity of these safety-critical events depends on the accuracy of the yielding behavior. Therefore, we will examine the yielding behavior simulated by NeuralNDE to further demonstrate its fidelity in modeling human interactions. The yielding behavior depends on the distance to the conflicting vehicle and the speed of the conflicting vehicle that is traveling within the roundabout. The results of yielding distance and yielding speed distributions are shown in Figure 3.12a and Figure 3.12b, respectively. We can find that NeuralNDE can perfectly replicate human yielding behavior and significantly outperforms the SUMO simulator. Human drivers are naturally heterogeneous and have different characteristics.

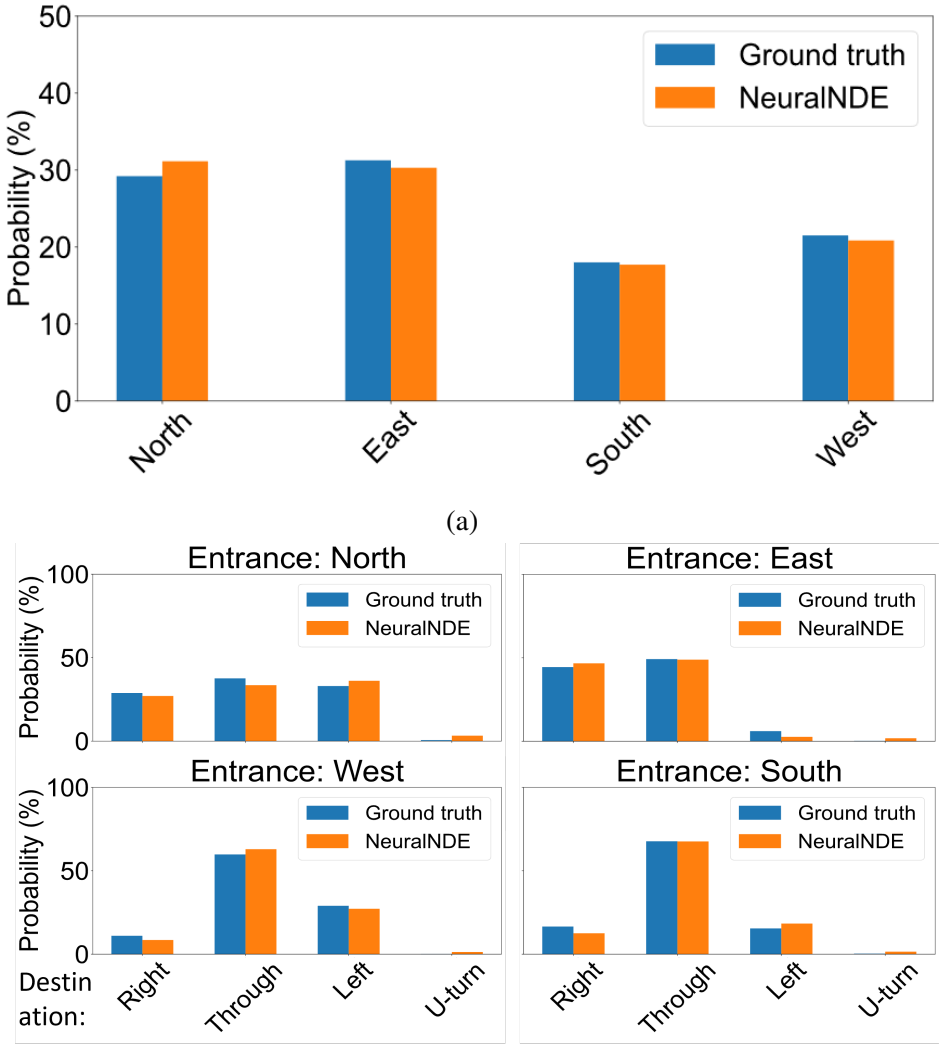
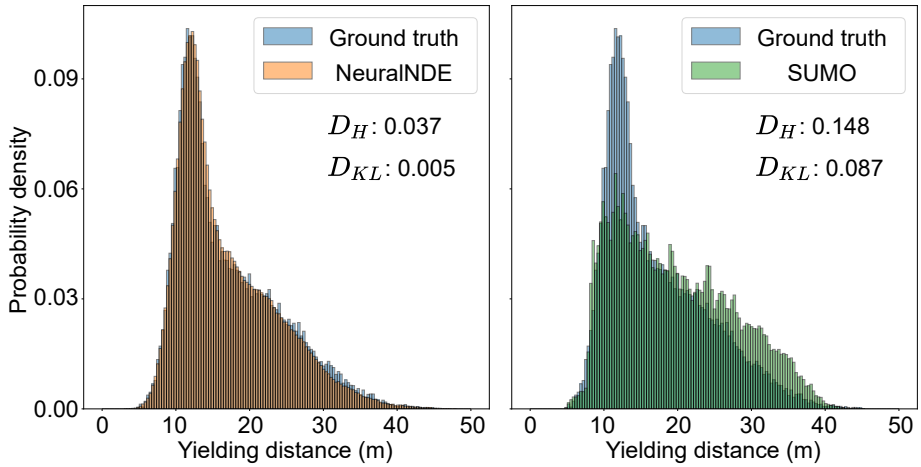


Figure 3.11: Statistical realism of normal driving behavior. (a) Traffic volume distribution. (b) Vehicle origin-destination (OD) distribution

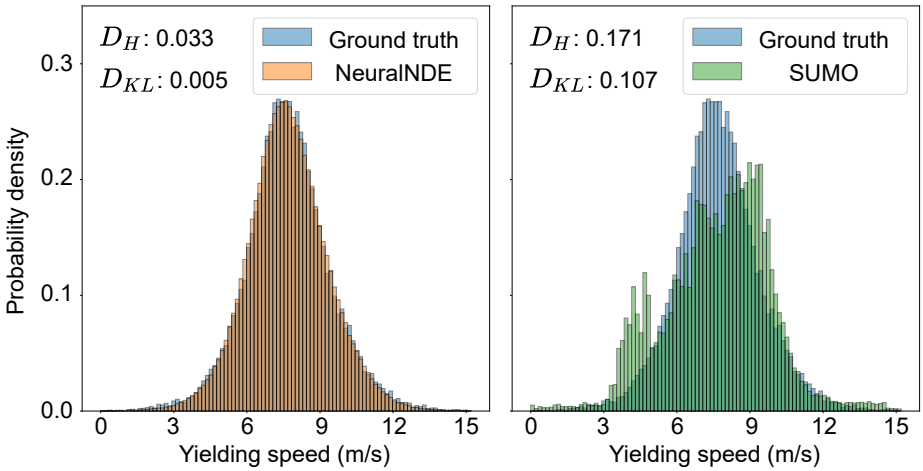
Different drivers often exhibit diverse driving behaviors and make different decisions, for example, some drivers are more aggressive and only give way when conflicting vehicles are very close while others might be more conservative. The proposed NeuralNDE is directly learned from real-world data without hand-crafted rules, therefore, it can master the nuanced yielding behavior of human drivers and generate a realistic and diverse driving environment.

3.4.4 Statistical realism of safety-critical driving behavior

The key challenge of current AV development is how to handle safety-critical driving situations occurring in the real world, therefore, the simulation environment must be able to reproduce



(a)



(b)

Figure 3.12: Statistical realism of normal driving behavior. (a) Yielding distance distribution: distance between the yielding vehicle and its nearest conflicting vehicle. (b) Yielding speed distribution: speed of the nearest conflicting vehicle

these long-tail rare events with high fidelity. In this section, we will examine the performance of NeuralNDE in generating safety-critical events, which include both crash and near-miss situations. The first important statistic is the crash rate. The Ann Arbor roundabout ground-truth crash rate is obtained based on data from August to mid-November 2021 for around 75 days from 7:00-19:00. There were 14 crashes in this roundabout with a total vehicle travel distance of 1.16×10^5 kilometers. Therefore, the empirical crash rate ground-truth is 1.21×10^{-4} crash/km. The crash rate of the NeuralNDE is 1.25×10^{-4} crash/km, which can accurately reproduce the real-world ground truth.

Not only can the proposed NeuralNDE reproduce an accurate crash rate, but also the detailed composition of crash types and crash severity distribution as shown in Figure 3.13a and Figure

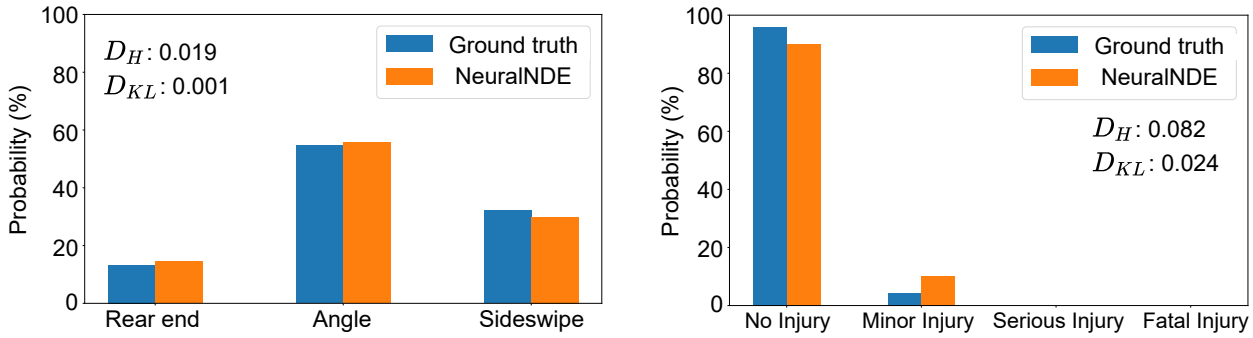
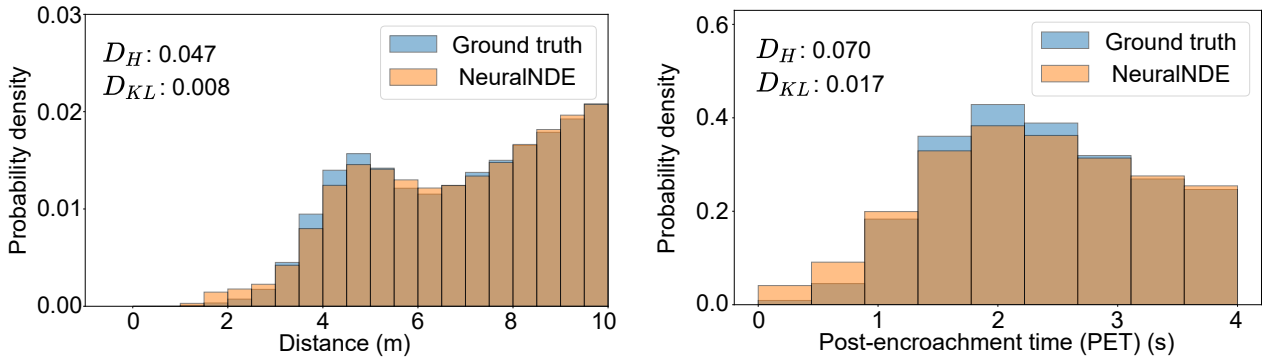


Figure 3.13: Statistical realism of safety-critical driving behavior. (a) Vehicle crash type distribution. (b) Vehicle crash severity distribution.

3.13b, respectively. The ground-truth crash type and crash severity distributions are queried from the Michigan Traffic Crash Facts dataset (Michigan Office of Highway Safety Planning, 2022) whose data is directly from police crash reports. We use data from 2016-2020, and there are a total of 520 crashes at this roundabout. For the crash severity, we use the worst injury of all involved occupants in the crash as the ground truth. Of the 520 crashes, 498 were non-injury crashes, 22 were minor injuries, and zero serious and fatal crashes. These demonstrate that NeuralNDE can generate accurate and diverse crash events following real-world occurring patterns, which are crucial for comprehensive testing of AV performance in different potential crashes. Compared with most state-of-the-art methods, for example, Refs (Suo et al., 2021; Bergamini et al., 2021; Kamenev et al., 2021; Igl et al., 2022; Meng et al., 2021), none of them compared their simulation results (e.g., crash rates/types/severities) against the real-world data. To the best of our knowledge, we are the only study that validated the simulated safety-critical statistics with real-world ground truth. For each crash type, we will further compare NeuralNDE-generated and real-world crash events in the later section to qualitatively demonstrate the fidelity of our approach. These results validate the capability and effectiveness of the proposed NeuralNDE in generating accurate crash statistics, which is critical for AV applications.

In addition to crashes, near-miss situations are also important. Two measurements, vehicle distance and PET distributions, are examined to validate the NeuralNDE fidelity. The closest distance between vehicles objectively characterizes potential conflicts between them. To validate the near-miss fidelity, we will focus on the vehicle distance that is smaller than a certain threshold, for example, 10 meters is used in this case. The PET is a widely used surrogate safety measure for identifying near-miss situations. The closer the distance and the smaller the PET, the more dangerous the situation. The results of the distance distribution in near-miss situations are shown in Figure 3.14a. We can find that NeuralNDE can replicate the distance in near-miss situations with



(a) (b)

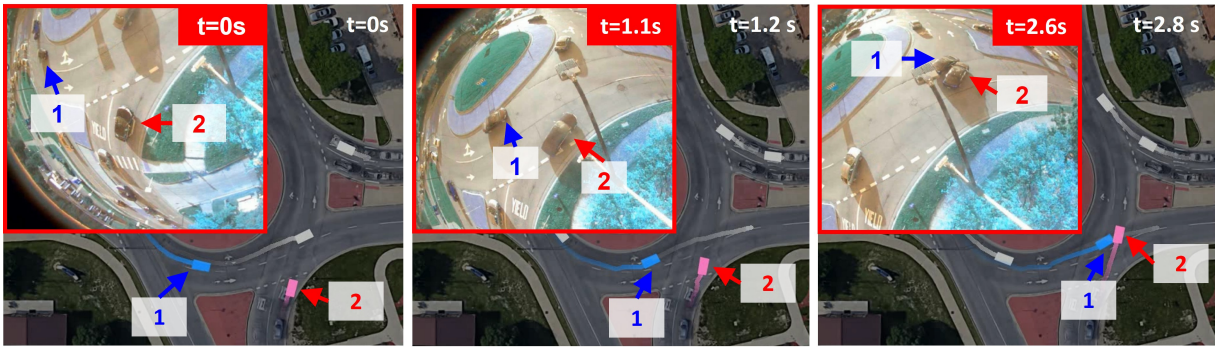
Figure 3.14: Statistical realism of safety-critical driving behavior. (a) Vehicle distance distribution in near-miss situations. (b) Post-encroachment time (PET) distribution in near-miss situations.

high accuracy. Similarly, the simulated PET distribution can also accurately reproduce real-world dangerous driving conditions as shown in Figure 3.14b. These results demonstrate that in addition to crashes, NeuralNDE can also characterize real-world near-miss statistics, which validates the modeling accuracy of the proposed method regarding vehicle safety-critical behaviors.

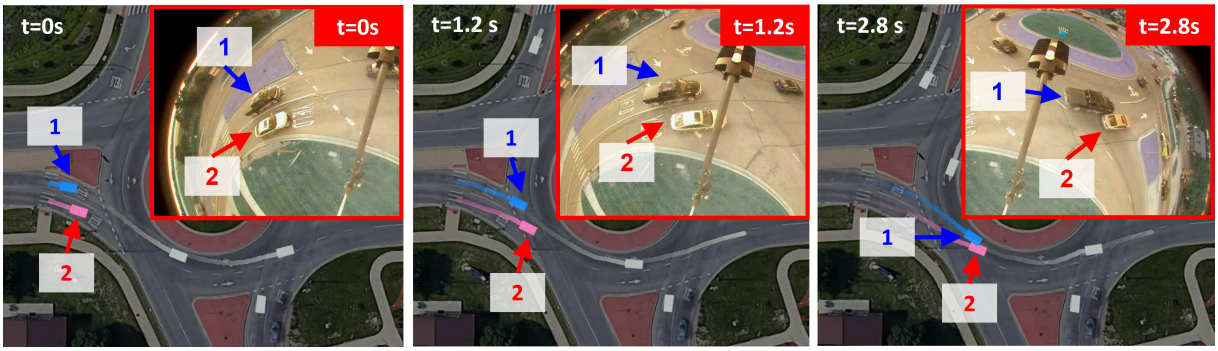
3.4.5 Generated crash events

The proposed NeuralNDE can generate complex and diverse interactions that happen in real-world traffic. During vehicle interactions, crashes may happen due to different reasons, for example, failure to yield, improper lane usage, etc. In this section, we showcase three generated crashes by NeuralNDE. By comparing them with real-world crash events, we can demonstrate that NeuralNDE can generate realistic and diverse crash patterns. These results further validate NeuralNDE fidelity on vehicle safety-critical behaviors which are very difficult to model. The illustration figures of the three crash examples with corresponding real-world crash events are shown in Figure 3.15.

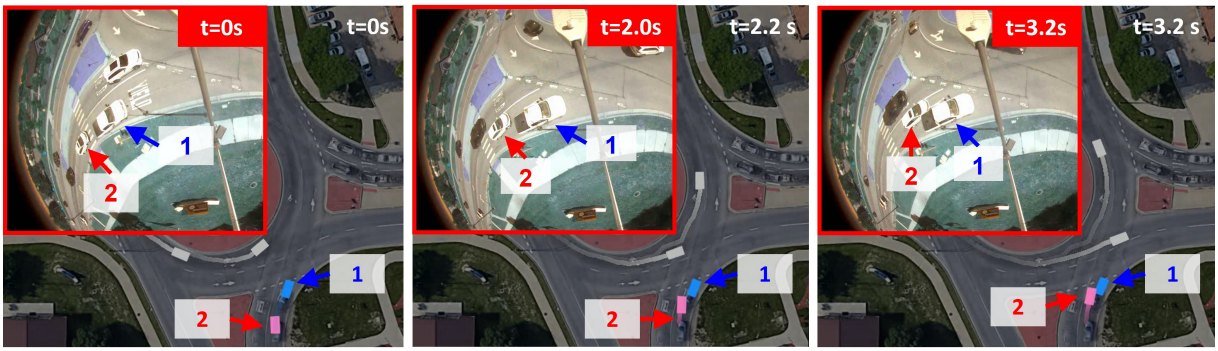
The first case is an angle crash caused by failure to yield as shown in Figure 3.15a, where the main image denotes the crash event generated by NeuralNDE, and the image in the red box is a real-world crash event. For the NeuralNDE results, vehicles' current states and their past trajectories are shown by rectangles and lines, respectively. For better visualization, only vehicles that are of our interest are shown in colors and other vehicles are shown in grey. In this case, vehicle $\#1$ (shown in blue) is circulating within the roundabout, and vehicle $\#2$ (shown in pink) is at the south entrance. We can find that vehicle $\#2$ fails to yield to the right-of-way of vehicle $\#1$, and chooses to enter the roundabout aggressively. As a result, vehicle $\#1$ cannot decelerate in time and a crash happens. The generated crash is very similar to what would happen in the real



(a)



(b)



(c)

Figure 3.15: Crash events in the real world and NeuralNDE. (a) Angle crash caused by failure to yield. (b) Sideswipe crash caused by improper lane usage. (c) Rear-end crash caused by failure to stop within assured clear distance.

Table 3.1: Ablation study results

	Backbone network	Safety Mapping Network	Adversarial Training	Conflict Critic Module	Instantaneous Speed ↓	Distance ↓	Yielding Distance ↓	Yielding Speed ↓	Crash Rate (crash/km)	Crash Type ↓
1	MLP	✓	✓	✓	0.119	0.076	0.194	0.073	7.81×10^{-4}	0.130
2	LSTM	✓	✓	✓	0.092	0.055	0.090	0.092	1.76×10^{-4}	0.113
3	Transformer	×	✓	N/A	**0.037**	**0.030**	0.036	0.033	2.34×10^{-1}	0.333
4	Transformer	✓	×	✓	0.038	0.032	**0.033**	0.040	1.29×10^{-4}	0.041
5	Transformer	✓	✓	×	0.040	0.032	0.036	**0.032**	1.22×10^{-5}	0.580
6	Transformer	✓	✓	✓	0.040	0.031	0.037	0.033	$\mathbf{1.25 \times 10^{-4}}$	**0.019**

N/A in the table means not applicable.

world as shown by the images in the red box of Figure 3.15a. As captured by the roadside camera, vehicle $\#2$ at the entrance fails to yield and finally crashed with vehicle $\#1$ within the roundabout.

The second case is a sideswipe crash caused by improper lane usage as shown in Figure 3.15b. In this case, two vehicles enter the roundabout from the west entrance side by side. Vehicle $\#1$ (shown in blue) drives in the inner lane and vehicle $\#2$ (shown in pink) drives in the outer lane. When they are approaching the south part of the roundabout, vehicle $\#2$ recklessly steers into vehicle $\#1$'s lane and leads to a crash. This type of improper lane usage crash also frequently occurs in the real world. As shown by the images in the red box of Figure 3.15b, vehicle $\#1$ also improperly intrude into the lane of vehicle $\#2$, causing a crash to happen.

The third case is a rear-end crash caused by failure to stop within assured clear distance. In this case, vehicle $\#1$ (shown in blue) is stopped and waiting to enter the roundabout, while vehicle $\#2$ (shown in pink) fails to maintain a safe distance from vehicle $\#1$ and causes a rear-end collision. The NeuralNDE-generated crash is very similar to the crash event happening in real traffic as shown in Figure 3.15c. From these results, we can find that NeuralNDE can generate realistic crash events that occur in the real world. The ability to reproduce these rare safety-critical events is essential for AV testing.

3.4.6 Ablation study

To examine the effectiveness of each module of the proposed framework, we conduct ablation studies in this section. Each model setup was run for approximately 10,000 hours of simulation. The results are shown in Table 3.1. Besides the crash rate column, other values in the table denote the Hellinger distance between the simulated distribution and ground-truth distribution. The smaller the Hellinger distance for the metrics, the better the model performance. For the crash rate, the closer to the real-world ground truth (1.21×10^{-4} crash/km), the better the performance.

To investigate the effects of the behavior network backbone, a two-layer MLP (hidden dimension equals 256) with batch normalization layers and Relu activation function is compared.

The main difference between the MLP and Transformer is that the Transformer utilizes the self-attention mechanism in its architectural design. By formulating the road agents as individual tokens, the self-attention mechanism in Transformer is naturally capable of characterizing inter-token interaction between agents. The result shows that the performance of the Transformer backbone is significantly better in all metrics compared to the MLP backbone. We also compared the Long Short-Term Memory (LSTM) network (two layers, hidden dimension equal to 256) as the backbone architecture. To model the interactions between agents, the LSTM module is embedded in a Seq-to-Seq framework (Sutskever et al., 2014) as a recurrent unit. This design allows the network to handle the interactions among all input agents instead of only historical ones. The results show that Transformer can achieve better performance in modeling vehicle interactions.

We also examine the importance of the safety mapping network. The conflict critic module is not applicable without the safety mapping network. From the results, we can find the crash rate is extremely unrealistic and multiple magnitudes higher than the ground truth. This result validates the performance of the proposed safety mapping network that significantly reduces the modeling error in safety-critical situations. The model exhibits good performance in other metrics since it does not consider safety performance and only optimizes to imitate normal driving behaviors.

Finally, we demonstrated the significance of the conflict critic module and the adversarial training. We cannot control the generation process of safety-critical events without the conflict critic module, therefore, we cannot obtain accurate crash rate and crash type distribution. Without adversarial training, we found that the crash rate and crash type distribution would be degraded. Table 3.1 demonstrates that the proposed model exhibits the overall best performance considering all evaluation metrics.

3.4.7 Model scalability

Modeling a large traffic network is more challenging than modeling individual scenarios because of two reasons: 1. It will be difficult to obtain full trajectory data for all vehicles in the network; 2. Error accumulation issue may become more noticeable because the elapsed time for each agent will be longer. The key idea for extending to a traffic network is that a large network can be decomposed into subareas, where critical subareas (e.g., intersection, roundabout, highway entrance and exit, etc.) that involve complex interactions will be controlled by NeuralNDE models, and other subareas (e.g., road segments connecting different scenarios, etc.) can be controlled by traditional rule-based models. Therefore, we only need to have trajectory data to build NeuralNDE models for those critical nodes in a large network, and connect these nodes with links that are modeled by traditional rule-based approaches (for example, car-following and lane changing models).

As a proof of concept, we build a "network", as shown in Figure 3.16, that involves two

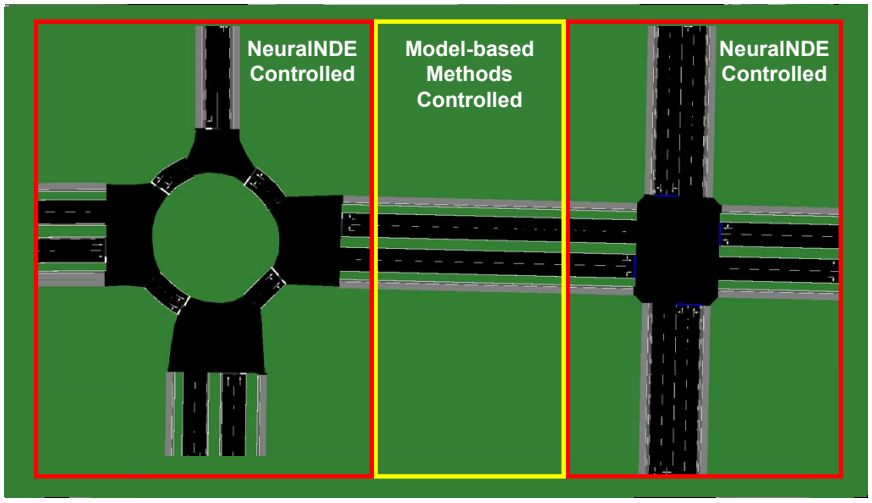


Figure 3.16: Proof-of-concept for modeling a road network

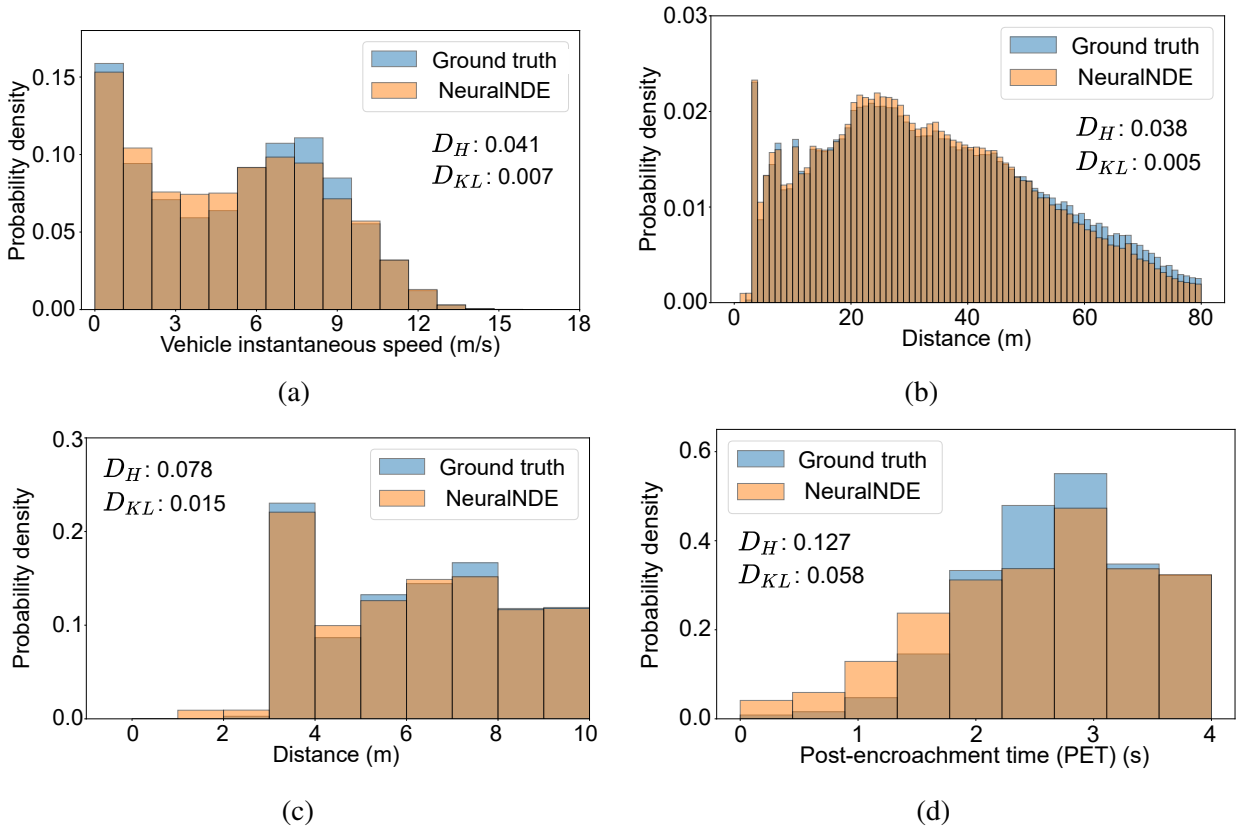


Figure 3.17: Statistical realism of the intersection area in the road network. (a) Vehicle instantaneous speed distribution. (b) Vehicle distance distribution. (c) Vehicle distance distribution in near-miss situations. (d) Post-encroachment time (PET) distribution in near-miss situations.

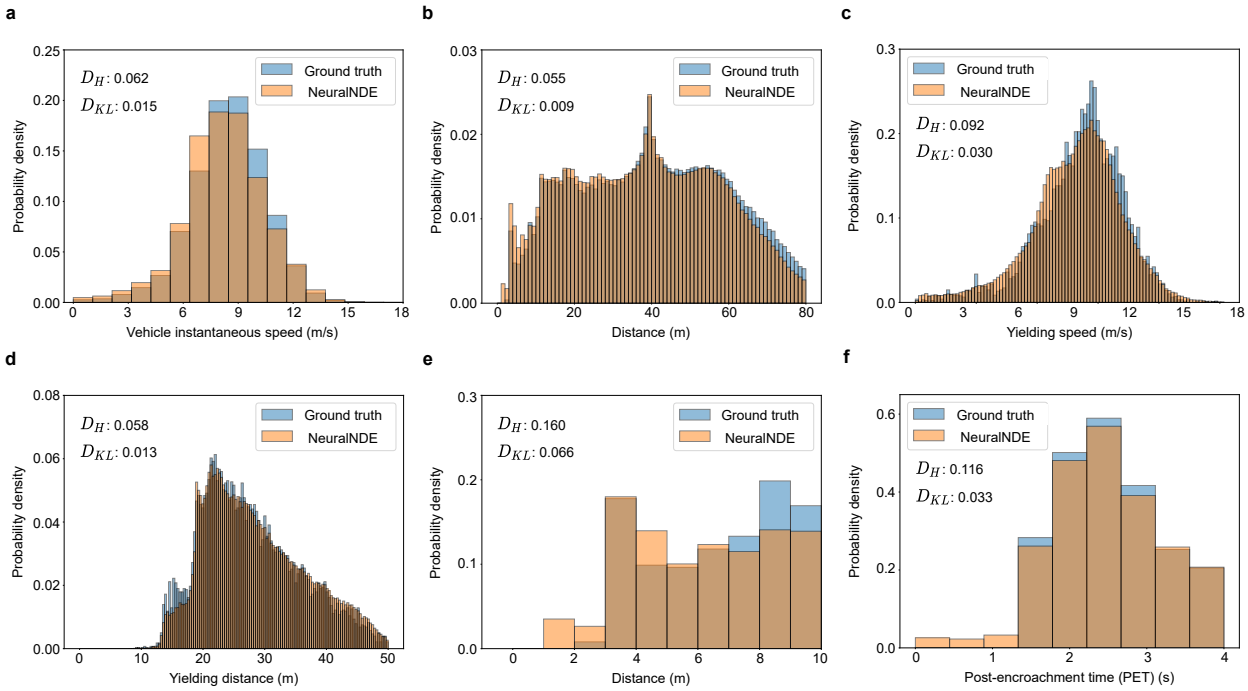


Figure 3.18: Statistical realism of the roundabout area in the road network. (a) Vehicle instantaneous speed distribution. (b) Vehicle distance distribution. (c) Yielding speed distribution. (d) Yielding distance distribution. (e) Vehicle distance distribution in near-miss situations. (f) Post-encroachment time (PET) distribution in near-miss situations.

scenarios, i.e., a four-way stop sign-controlled intersection and a two-lane roundabout. We use SUMO (Lopez et al., 2018) simulator to generate vehicle trajectory data and use it as the ground truth of the NDE. We assume the traffic network is not fully perceptional and we only have vehicle trajectory data in the intersection and roundabout areas. Therefore, these two areas (shown by red rectangles in Figure 3.16) are controlled by trained NeuralNDE models, and the transition areas ((shown by the yellow rectangle in Figure 3.16)) between the two scenarios are controlled by rule-based IDM car-following model (Treiber et al., 2000) and SL2015 lane-changing model. More details about experiment settings can be found in the Appendix B.4.

We simulate the network and collect the data in intersection and roundabout areas to quantitively evaluate the performance. The simulated network can still achieve statistical realism and the results are discussed below. We ran around 100 hours of simulation to collect the data. For the intersection area, we evaluate vehicle instantaneous speed and distance distributions to demonstrate the performance of normal driving behavior, as shown in Figure 3.17a-b, respectively. From the results, we can find that the simulated distribution is consistent with the ground truth. We further validate the statistical realism of the safety-critical driving behavior. The results of vehicle distance in near-miss situations (smaller than 10 meters) and PET are shown in Figure 3.17c-d, respectively. We can find that the proposed method can replicate the ground-truth with

high accuracy.

For the roundabout area, the vehicle instantaneous speed, distance, yielding speed, and yielding distance results are shown in Figure 3.18a-d, respectively. The safety-critical events metrics (vehicle distance in the near-miss situations and PET) are shown in Figure 3.18e-f, which also demonstrate satisfactory performance. These results serve as a proof of concept to demonstrate the performance and scalability potential of our proposed NeuralNDE models for simulating large traffic networks.

3.5 Summary

In this chapter, we proposed NeuralNDE, a deep learning-based NDE modeling framework. The proposed NeuralNDE demonstrates promising performance for modeling real-world complex urban driving environments with statistical realism for both normal and safety-critical driving conditions. Compared with existing works, this is the first time that a simulation environment can statistically reproduce real-world driving environments with such high accuracy. More importantly, it can accurately characterize long-tail rare-event statistics, for example, crash rate, crash type, and crash severity distributions, which are very difficult to achieve but will notably influence AV training and testing accuracy. The proposed NeuralNDE model focuses on modeling the microscopic behavior of human drivers, addressing a significant gap in mainstream AV simulators (e.g., CARLA (Dosovitskiy et al., 2017) that predominantly emphasize photorealistic rendering and sensor simulations. Therefore, the NeuralNDE can be directly integrated with them to constitute a complete simulation suite.

Beyond its applications in the context of AVs, NeuralNDE boasts wide-ranging potential across diverse domains. For instance, it can serve as a tool for assessing the safety performance of traffic facilities under varying traffic flow conditions. In essence, this high-fidelity microscopic simulator equips us to address 'what-if' questions within the realm of transportation engineering.

Despite the promising potential of the proposed method, it still comes with specific limitations and can be improved in the future. Firstly, incorporating road geometry information into the model input could further enhance its generalizability. Secondly, as human-driven vehicles may exhibit distinct behaviors when interacting with AVs, further development might be required to consider the AV influences on surrounding human-driven vehicles.

CHAPTER 4

Evaluate AV Safety Performance using NDE Model

4.1 Introduction

4.1.1 Background and related works

One important application of the NDE is to test AV safety performance. There are two critical problems that need to be solved for AV testing. The first one is how to build the high-fidelity NDE, which determines the trustworthiness of testing results. Our works discussed in Chapter 2 and 3 are focusing on this problem. However, it is not efficient to directly evaluate AV in NDE, since it has been argued that hundreds of millions of miles are required to demonstrate the AV safety performance at the level of human drivers (Kalra and Paddock, 2016). Therefore, the second problem is how to develop testing algorithms to improve testing efficiency.

Please refer to Section 1.3.2 for a more detailed literature review on testing methodologies to improve testing efficiency. Some representative works include Zhao et al. (2016); O'Kelly et al. (2018); Feng et al. (2020d,a). The core concept revolves around evaluating AV in more challenging scenarios created using IS-based techniques. By increasing the occurrence probability of safety-critical events, these methods can improve the testing efficiency in isolated scenarios, such as the cut-in scenario. However, it's important to note that most of these scenario-based approaches are only suited for short-duration segments, typically lasting only a few seconds and involving a limited number of background road users, usually one or two BVs. The ultimate objective, however, is to assess AV performance within spatial-temporally continuous NDE, which encompasses a multitude of background road users and a long time horizon. Furthermore, it should be noted that evaluating AV performance solely in isolated scenarios may not offer a comprehensive assessment.

4.1.2 Overview of the chapter

The focus of this chapter is to develop a method for evaluating the safety performance of AV. To achieve evaluation efficiency without loss of accuracy, the proposed approach is based on NDE developed in Chapter 2, but with sparse but intelligent adjustments, resulting in the NADE. The NADE is both naturalistic and adversarial, in that most of the BVs (more generally, road users) follow naturalistic behaviors for most of the time, and only at selected moments, selected vehicles execute specific designed adversarial moves. By training the BVs to learn when to execute what adversarial maneuver, the proposed NADE becomes an intelligent environment for AV testing.

We demonstrate the effectiveness of the proposed method in the highway-driving environment. Two representative AV agents based on model-based and Deep Reinforcement Learning (DRL) methods are developed. Simulation results show that the NADE could significantly accelerate the evaluation process by multiple orders of magnitude with the same accuracy, comparing with the NDE-based method.

4.1.3 Contributions and organization of the chapter

The contributions of this chapter are summarized as follows:

1. We propose a AV testing method that provides spatiotemporally continuous testing environments, which enables comprehensive evaluation of AV safety performance.

2. The proposed method ensures the testing results (such as accident rates of different accident types) of AVs in the generated environment are unbiased with the NDE.

3. The proposed method addresses the inefficiency issue of AV testing. Compared with testing in NDE, the proposed method reduces the testing time by multiple orders of magnitude for the same evaluation accuracy.

This chapter is organized as follows: Section 4.2 introduces the overall framework of the proposed method and discusses the detailed methodology of NADE generation and the evaluation of AVs with NADE. Section 4.3 is a case study demonstrating the performance of the proposed method. Section 4.4 provides a summary of this chapter.

4.2 Methodology

4.2.1 Overall framework

To test AV accurately and efficiently, we design the testing environment based on NDE with sparse but intelligent adjustments, resulting in NADE, to improve the occurrence probability of safety-

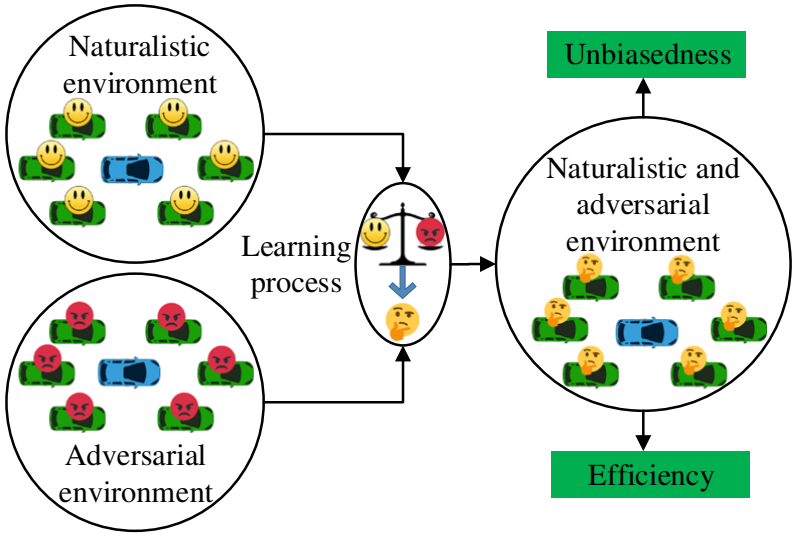


Figure 4.1: The NADE learns to balance the naturalistic environment and adversarial environment for AV testing based on the agent-environment framework, while ensuring unbiasedness and improving efficiency

critical events, which rare to occur in NDE. As shown in Figure 4.1, the key to creating the NADE is to train the background vehicles in the NDE to learn when to execute what adversarial maneuver while ensuring unbiasedness and improving the efficiency. The learning process is guided by our theoretical discovery below.

In essence, AV testing can be considered as a rare event estimation problem with high-dimensional variables. However, few existing methods can handle both the challenges of the rareness of events and high dimensionality. Testing AVs in NDE is an application of the MC theory (Mooney, 1997), which suffers from inefficiency problems for rare events. The IS theory has been developed for solving the challenge of rare events, but it can only be applied in low-dimension situations (Au and Beck, 2003). It was proved that its efficiency would decrease exponentially with the increase of dimensionality. Therefore, both MC and IS have limitations for the rare event estimation problem with high-dimensional variables. However, people have not paid much attention to the advantage of the MC theory for high dimensionality. We discover that, if there exists a small subset of variables that are critical to the rare events, applying IS theory with the small subset of variables while applying the MC theory with the remaining variables can help overcome both the challenges of the rareness of events and high dimensionality. We provide a theoretical proof of this in Section 4.2.5. This is significant as this can apply to a general set of problems with such characteristics. For safety-critical performance tests of AVs, fortunately, these small but critical variables exist because most of the vehicle accidents involve only a small number of vehicles in a short period (Swanson et al., 2019). According to the Fatality Analysis Reporting

System (FARS), about 91.5% fatal injuries suffered in motor vehicle traffic crashes in the United States in 2018 involved only one or two vehicles (United States. Department of Transportation, 2018).

4.2.2 Generation of NDE

To test AV accurately and efficiently, we design the testing environment based on NDE with sparse but intelligent adjustments, resulting in NADE, to improve the occurrence probability of safety-critical events while ensuring unbiasedness. The NDE can be represented as

$$x = \begin{bmatrix} x_{1,1} & \dots & x_{1,T} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \dots & x_{N,T} \end{bmatrix}, x \in X, \tag{4.1}$$

where $x_{i,j}$ denotes the variables (e.g., position and speed) of the i-th BV at the j-th time step, N denotes the number of BVs of interest, T denotes the total number of time steps, and X denotes the feasible space of variables. The NDE generation is to sample values of variables according to their naturalistic joint distributions, denoted as $x\ P(X)$.

As $P(X)$ is extremely high dimensional, we simplify the problem by exploiting spatiotemporal independence among the variables. Assuming the Markovian property, the joint distribution can be simplified in a factorized way as

$$P(x) = P\left(s\left(0\right)\right) \times \prod_{k=0}^{T} P\left(u\left(k\right)|s\left(k\right)\right). \tag{4.2}$$

Here, the state and action at the time step $k = 0, ..., T$ are denoted as

$$\begin{aligned} s\left(k\right) &= \left[s_0(k), s_1(k), \dots, s_N(k)\right], \\ u\left(k\right) &= \left[u_1(k), \dots, u_N(k)\right], \end{aligned} \tag{4.3}$$

where s_0 denotes the state of the AV under test, $s_i (i = 1, ..., N)$ denotes the state of the i-th BV, and u_i denotes the maneuver (e.g., longitudinal accelerations) of the i-th BV. Then the NDE is generated by sampling maneuvers as $u(k)\ P(u(k)|s(k)$ at each time step. To simplify $P(u(k)|s(k)$, it is assumed that all BVs choose their maneuvers simultaneously and independently, so we can calculate it in a factorized way as

$$P\left(u\left(k\right)|s\left(k\right)\right) = \prod_{i=1}^{N} P\left(u_i\left(k\right)|s\left(k\right)\right). \tag{4.4}$$

The action distribution $P\left(u\left(k\right)|s\left(k\right)\right)$ of each vehicle can be obtained based on results of Chapter 2, in which car-following and lane-changing behavior models are developed based on real-world data.

4.2.3 Generation of NADE

This section describes our algorithm for NADE construction. The key is to obtain new behavioral distributions $q\left(u|s\right)$ as the replacement of $P\left(u|s\right)$ in NDE. To overcome the challenges of high dimensionality, we identify the Principal Other Vehicle (POV) at the critical moment and only adjust its behaviors.

To identify the POV, we define the maneuver criticality as the multiplication of exposure frequency $P(u_i|s)$ and maneuver challenge $P(A_i|s,u_i)$ as

$$V(u_i|s) \triangleq P(u_i|s) \times P(A_i|s,u_i), \tag{4.5}$$

where A_i denotes the accident between the i-th BV and the AV under test. The first part on the right-hand side is the exposure frequency obtained from NDD. The second part is the maneuver challenge that indicates the accident probability given the state-action pair (s,u_i). Since we treat the AV model under test as a black box, to approximate the maneuver challenge, we construct Surrogate Model (SM)s of AVs by meta-models, described in more detail in Appendix C.1. We should note that the SMs can also be constructed based on the preliminary AV models, so it provides an elegant way to leverage existing testing results of preliminary AV models. Let S_i denote the accident between the i-th BV and the SMs. Then, the maneuver challenge can be approximated by

$$\begin{aligned} P\left(A_i|s,u_i\right) &= \sum_{u_0} P(u_0|s)P(A_i|s,u_i,u_0), \\ &\approx \sum_{u_0} P(u_0|s)P(S_i|s,u_i,u_0), \end{aligned} \tag{4.6}$$

where $P(u_0|s)$ denotes the probability of the AV's maneuver u_0 at the state s. The first term $P(u_0|s)$ can be predicted approximately by the SMs, and the second term $P(S_i|s,u_i,u_0)$ can be evaluated by simulations of the SMs. Realizing that the evaluation of $P(S_i|s,u_i,u_0)$ may not be completed by one-time-step simulation, to obtain the evaluation result quickly, Reinforcement Learning (RL) or DRL methods may be used. In this chapter, we adopted RL techniques for the basic scenarios such as car-following, while more general scenarios can be approximated by the combination of basic scenarios. More details of the maneuver challenge calculation can be found in Section 4.3.1.

The criticality for each background vehicle can then be calculated as the summation of maneuver criticality over all BV's maneuvers:

$$C_i(s) \triangleq \sum_{u_i} V(u_i|s), \tag{4.7}$$

and the POV can be identified by

$$c \triangleq \arg\max_i C_i(s), \tag{4.8}$$

if $C_c(s) > C$, where C is a pre-determined threshold (e.g., 0). We define the moment as the critical moment if there is at least one POV. Because most accidents involve only two vehicles, we considered at most one POV at each moment in this chapter. The generalization of this work to multiple POVs is straightforward.

Finally, we construct the importance function by adjusting the maneuvers of POV at critical moment as

$$q(u|s) = q(u_s|s) \times \prod_{i=1, i\neq c}^{N} P(u_i|s), \tag{4.9}$$

where u_c denotes the maneuver of POV. Only the POV's maneuver is adjusted by $q(u_c|s)$, while other vehicles follow their naturalistic distributions as in NDE. For uncritical moments, all vehicles behave as in NDE. The $q(u_C|s)$ is constructed by the weighted average of the naturalistic distribution and the normalized criticality distribution as

$$q(u_c|s) = \epsilon P(u_c|s) + (1-\epsilon)\frac{V(u_c|s)}{C_c(s)}, \tag{4.10}$$

where $\epsilon > 0$ is the weight of the naturalistic distribution. It can balance the exploitation and exploration to mitigate the influence of approximation errors of maneuver criticality.

4.2.4 Evaluation of AVs with NADE

This section describes how to estimate the accident rate of AV when testing with NADE. Specifically, if the event of interest (accident event of AVs in this paper) is denoted as A, we can measure the driving intelligence of AVs by

$$P(A) = \sum_{x\in X} P(A|x)P(X), \tag{4.11}$$

where x denotes variables of the driving environment, and X denotes its feasible domain. The NDE-based testing method is essentially to estimate $P(A)$ by the MC method as

$$\begin{aligned} P\left(A\right) &\approx \frac{1}{n}\sum_{i=1}^{n} P(A|x_i), x\ P(x), \\ &\approx \frac{m}{n}, \end{aligned} \tag{4.12}$$

where n denotes the number of tests, m denotes the number of the event A during the tests, and $x\ P(X)$ indicates that the variables are sampled from their naturalistic distributions.

Because the event $A$is usually a rare event for AVs in NDE, the MC method suffers from severe inefficiency limitations. To mitigate this issue, the IS method was applied for scenario-based methods as

$$\begin{aligned} P\left(A\right) &= \sum_{x\in X} \frac{P(A|x)P(X)}{q(x)} q(x), \\ &\approx \frac{1}{n}\sum_{i=1}^{n} \frac{P(A|x_i)P(X_i)}{q(x_i)}, x_i \sim q(x), \end{aligned} \tag{4.13}$$

where $q(x)$ is called the importance function. By introducing importance functions, the testing priority of critical scenarios will be improved, and so will the evaluation efficiency (Feng et al., 2020d,a,c,b). However, all existing IS-based methods suffer from the "curse of dimensionality" (Au and Beck, 2003), and thus cannot be applied directly for the complex driving environment.

We solve the "curse of dimensionality" by combining MC and IS methods. Conceptually, only the critical variables are adjusted by the IS method, while other variables keep their naturalistic distributions following the MC method. Following the formulation and assumptions in NDE and NADE, we derive the performance estimation equation as

$$P(A) \approx \frac{1}{n}\sum_{i=1}^{n} \left(P\left(A|x_i\right) \times \left[\prod_{k=1}^{T_i} \frac{P(u(k)|s(k))}{q(u(k)|s(k))} \right] \right), \tag{4.14}$$

where T_i denotes the total time steps of the i-th simulation test. In this chapter, we terminate a test if an event A happens or the test reaches the pre-determined driving distance. Denote $T_{i,c}$ as the set of critical moments of the i-th test, and, finally, the performance estimation equation can be obtained as

$$P(A) \approx \frac{1}{n}\sum_{i=1}^{n} \left(P\left(A|x_i\right) \times \left[\prod_{k\in T_{i,c}} R(k) \right] \right), \tag{4.15}$$

where

$$R(k) \triangleq \prod_{k=1}^{T_i} \frac{P(u(k)|s(k))}{q(u(k)|s(k))}, \tag{4.16}$$

is the simulation weight (likelihood ratio) recorded during the test process. The $P(A|x_i)$ is estimated by counting the number of accident events occurring in the test. Based on this equation, the accident rate of the AV under test can be estimated by the testing results in NADE.

4.2.5 Theoretical analysis of accuracy and efficiency of NADE

This section theoretically justifies the accuracy and efficiency of our NADE-based testing method. As proved by the importance sampling theory (Owen, 2013), the performance evaluation is unbiased if $q(x) > 0$ whenever $P(A|x)P(x) \neq 0$. As $\epsilon > 0$ in the generation of NADE, we can guarantee $q(u|s) > 0$ whenever $P(u|s) \neq 0$ for all states and actions, which is sufficient for the unbiasedness. Therefore, our NADE-based testing method is statistically accurate.

To justify the efficiency of our method, we introduce the lemma regarding the "curse of dimensionality" of the IS method (Au and Beck, 2003)

Lemma 4.1. *The estimation variance of the IS method has the lower bound as:*

$$\sigma^2 \geq P^2(A) \left\{ \exp \left[D_{KL} \left(q^\star(x) \| q(x) \right) \right] - 1 \right\}, \tag{4.17}$$

where $q^\star(x)$ is the optimal importance function with zero estimation variance, and

$$D_{KL} \left(q^\star(x) \| q(x) \right) = E_{q^\star(x)} \left(\log \frac{q^\star(x)}{q(x)} \right), \tag{4.18}$$

is the KL-divergence as the measurement of discrepancies between $q^\star(x)$ and $q(x)$.

Following the independence assumptions in NDE, if the IS method is directly applied, we can derive the equations as

$$q^\star(x) = q^\star \left(s(0) \right) \times \prod_{k=1}^{T} q^\star \left(u(k)|s(k) \right), \tag{4.19}$$

$$\log \frac{q^\star(x)}{q(x)} = \log \frac{q^\star \left(s(0) \right)}{q \left(s(0) \right)} + \sum_{k=1}^{T} \sum_{i=1}^{N} \log \frac{q^\star \left(u_i(k)|s(k) \right)}{q \left(u_i(k)|s(k) \right)}. \tag{4.20}$$

As $\log \frac{q^\star(u_i(k)|s(k))}{q(u_i(k)|s(k))}$ is usually predetermined by prior knowledge, the KL-divergence will increase linearly with the dimensionality $O\left(NT\right)$, and, therefore, the estimation variance will increase exponentially with the dimensionality, leading to the "curse of dimensionality".

For NADE, if the variance is only dependent on the dimensionality of the adjusted critical variables, i.e., the maneuvers of POV at the critical moments, then our method addresses the "curse of dimensionality". Specifically, if we denote x_c the critical variables, which are independent of all other variables x_{-c}, we propose the theorem as follows, and the proof can be found in Appendix C.3.

Theorem 4.1. *The estimation variance of our method has the following relations:*

$$\begin{aligned}\sigma^2 &= P^2(A) D_{\chi^2}\left(q^\star(x_c) \| q(x_c)\right) + D\left(x_c \| x\right), \\ &\geq P^2(A)\left\{\exp\left[D_{KL}\left(q^\star(x_c) \| q(x_c)\right)\right] - 1\right\} + D(x_c \| x),\end{aligned} \tag{4.21}$$

where $D_{\chi^2}\left(q^\star(x_c) \| q(x_c)\right) = E_{q(x_c)}\left(\left(\frac{q^\star(x_c)}{q(x_c)} - 1\right)^2\right)$ denotes the χ^2-divergence, $q^\star(x_c) = \frac{P(A|x_c)P(x_c)}{P(A)}$ denotes the optimal importance function for the critical variables, and $D(x_c \| x) = E_{q(x)}\left[\left(P(A|x) - P(A|x_c)\right)^2 \frac{P^2(x)}{q^2(x)}\right]$ measures whether the adjusted variables are critical.

The term $D(x_c \| x)$ measures the variance caused by the identification of critical variables. The more critical the adjusted variables x_c are, the closer $P(A|x_c)$ is to $P(A|x)$, and thus the closer $D(x_c \| x)$ is to zero.

The KL-divergence and χ^2-divergence measure the discrepancies between optimal importance functions and proposed importance functions. Compared with Lemma 4.1, both the divergences are related to the dimensionality of the critical variables, instead of all variables, which resolves the challenge of high dimensionality for rare event estimation problems.

4.3 Case studies

4.3.1 Generation of NADE

We demonstrated the effectiveness of our method for AV testing in a highway driving environment. The NDE used in this chapter is developed based on the car-following and lane-changing models discussed in Chapter 2. At each moment, each BV has 33 possible maneuvers: left lane change, 31 car following accelerations (range from -4 to $2m/s^2$ with a resolution of 0.2, and right lane change. All vehicles are assumed to select maneuvers independently and simultaneously for each time step. For simplification, the action is updated every 1 second (one time step). The underlying highway traffic simulator (Leurent, 2018) determines specific positions, speeds, and steering angles of all vehicles with bicycle models at a frequency of 15 Hz during each time step. All lane-changing maneuvers are set completed within one time step. The simulation continues until all simulation time steps are completed or a crash event happens.

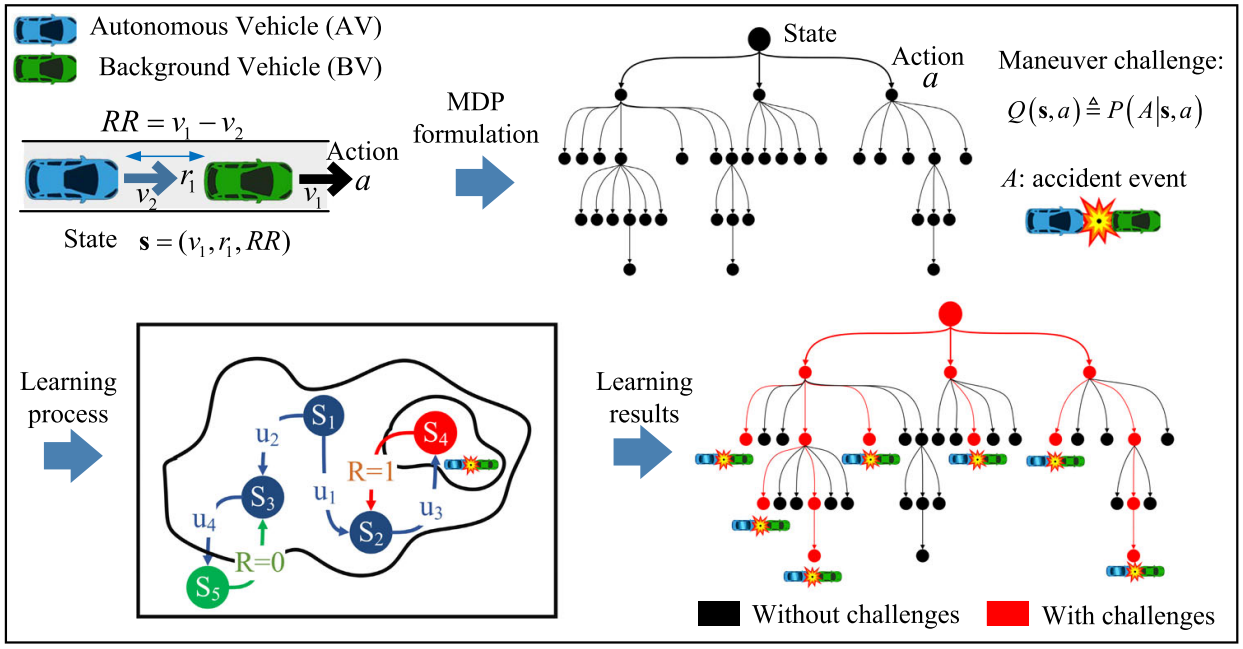


Figure 4.2: Maneuver challenge calculation of the BV's accelerations for car following scenarios based on reinforcement learning techniques

The most significant part of our method is the generation of NADE for AV testing. In essence, we aim to construct new distributions, as the replacement of the naturalistic behavior distributions in NDE, for sampling maneuvers of BVs. To solve the challenge of high dimensionality, we only twist the behavior distributions of the POV at critical moments. To identify the POV, at each time step, each BV's maneuver is evaluated by the criticality metric, which can be computed as a multiplication of exposure frequency and maneuver challenge. The exposure frequency represents the naturalistic probability of the maneuver in NDE, while the maneuver challenge measures its safety challenge to the AV under test.

Therefore, one important step of our method is to calculate the maneuver challenge of each BV's maneuver at every state. As the calculation of maneuver challenge involves the interdependency of maneuvers from both the AV and BVs in the following time steps, the RL method is adopted for basic scenarios such as car-following, while more general scenarios can be approximated by the combination of basic scenarios. As the specifics of behavior model of the AV under test are usually unknown, we utilize SMs to approximate the maneuver challenge. Although approximation errors usually exist, the maneuver challenge can provide valuable information on the impact of BV's maneuvers. SMs can be constructed based on common knowledge of AVs or prior tests of AVs. In this chapter, we utilize the IDM and MOBIL models as SMs, which are commonly used in the transportation domain. To capture the uncertainty of AVs, we modify the MOBIL model as a stochastic lane-changing model described in more detail in Appendix C.1.

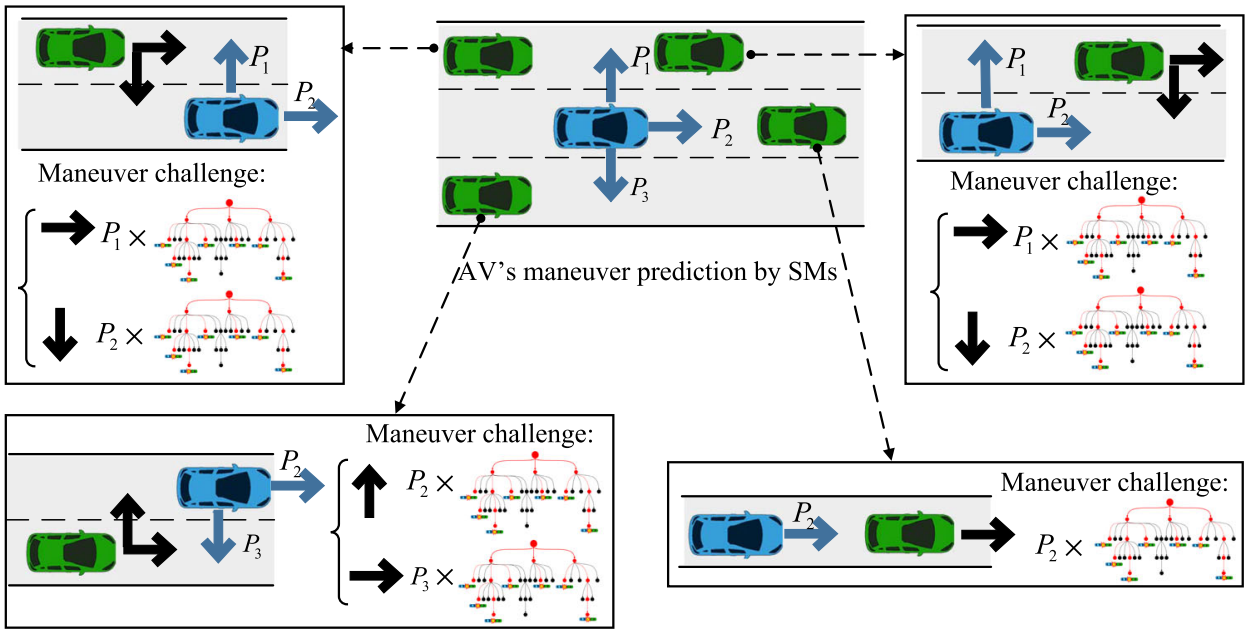


Figure 4.3: Example of maneuver challenge calculation for general scenarios

With SMs, we propose to learn the maneuver challenge for car-following scenarios by the RL method as illustrated in Figure 4.2. Specifically, the state is defined as the BV's speed, range, and speed difference, and the action is defined as the BV's acceleration. Based on Markov Decision Process (MDP), car-following scenarios can be represented by a decision tree, where each branch from the initial state to the terminal state specifies a car-following trajectory. To handle the delayed reward of AV's accidents, the state-action value of RL is defined as the maneuver challenge, while the reward is set to one for the AV's accident event and zero for safe states. The states and actions, which may eventually lead to accidents of the AVs, have positive challenge values. More technical details can be found in Feng et al. (2020a). The learning process took about 20 minutes to the convergence in a desktop computer equipped with Intel i7-7770 CPU and 16 GB RAM.

For general scenarios, we propose to calculate the maneuver challenge for each BV based on the maneuver prediction of the AV and the results of car-following scenarios. The basic idea is to calculate the maneuver challenge of each BV at the current time by taking the expectation of its maneuver challenge over all of its possible maneuvers at the next time step. The AV's maneuvers are predicted as a probability distribution by SMs. To demonstrate the computation of the maneuver challenge, let us take the BV in the top left of Figure 4.3 as an example. For the BV, there are two possible maneuvers, one is longitudinal acceleration, and the other is to take the right lane change. For the AV, there are three possible maneuvers, left lane change, longitudinal accelerations, and right lane change. Each of the maneuvers is predicted by the SM with a probability. Between the AV and the BV, there are a total of six possible maneuver

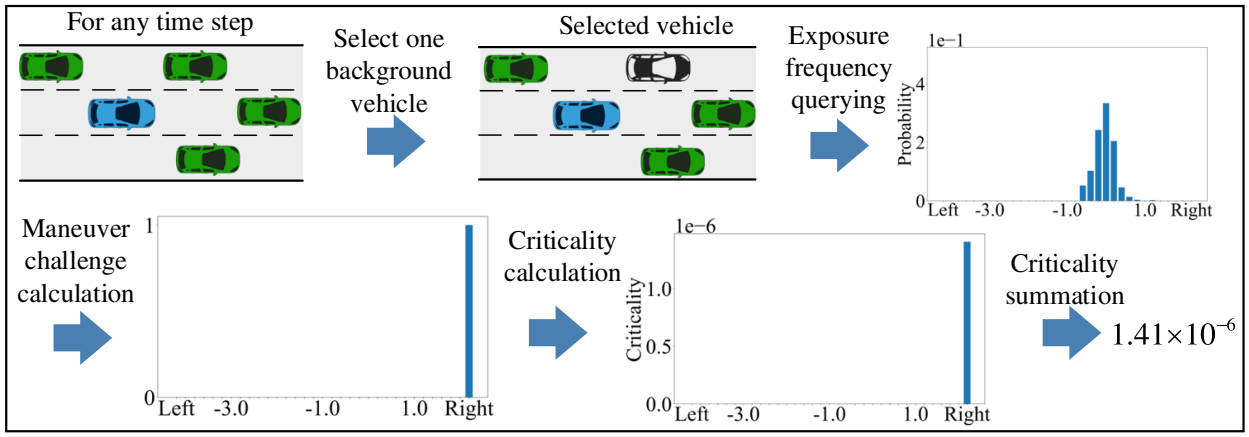


Figure 4.4: Example of criticality calculation

combinations, among which two of them are predicted to have non-zero maneuver challenges in the next time step. One is the BV makes a right lane change while the AV remains longitudinal, and the other is the BV remains longitudinal while the AV makes a left lane change. In both scenarios, the BV and the AV are in a car-following situation after the lane change maneuver, where the maneuver challenge can be obtained with the RL model discussed above. The overall maneuver challenge of the BV is an expectation of those in the two car-following situations.

After calculating the maneuver challenge, the criticality of each BV's maneuver at each state can be calculated. For example, as shown in Figure 4.4, the exposure frequency of each BV can be queried as in NDE, and the maneuver challenge is calculated as discussed above. Then the criticality is obtained by multiplying the exposure frequency and maneuver challenge. The criticality of most BVs' maneuvers is zero because either the exposure frequency is zero (impossible maneuver) or the maneuver challenge is zero (unchallenging maneuver).

Among all the BVs surrounding the AV, a BV is identified as the POV if its criticality value is the largest and larger than a threshold (0 in this chapter). The moment with a POV is identified as the critical moment. For the POV at the critical moment, the importance functions are constructed by the weighted average of the exposure frequency and the normalized criticality: with the probability ϵ, we sample maneuvers from the exposure frequency, while with the probability $1-\epsilon$, we sample maneuvers from the normalized criticality. Inspired by the defensive importance sampling, the weighted average can mitigate the influences of the approximation errors of maneuver challenge. The maneuver of POV at the critical moment is then sampled from the importance function, while maneuvers for all other vehicles are sampled from the naturalistic distribution as in NDE. This completes the simulation for one time step with all vehicle states updated. Figure 4.5 shows an example of the NADE generation procedure.

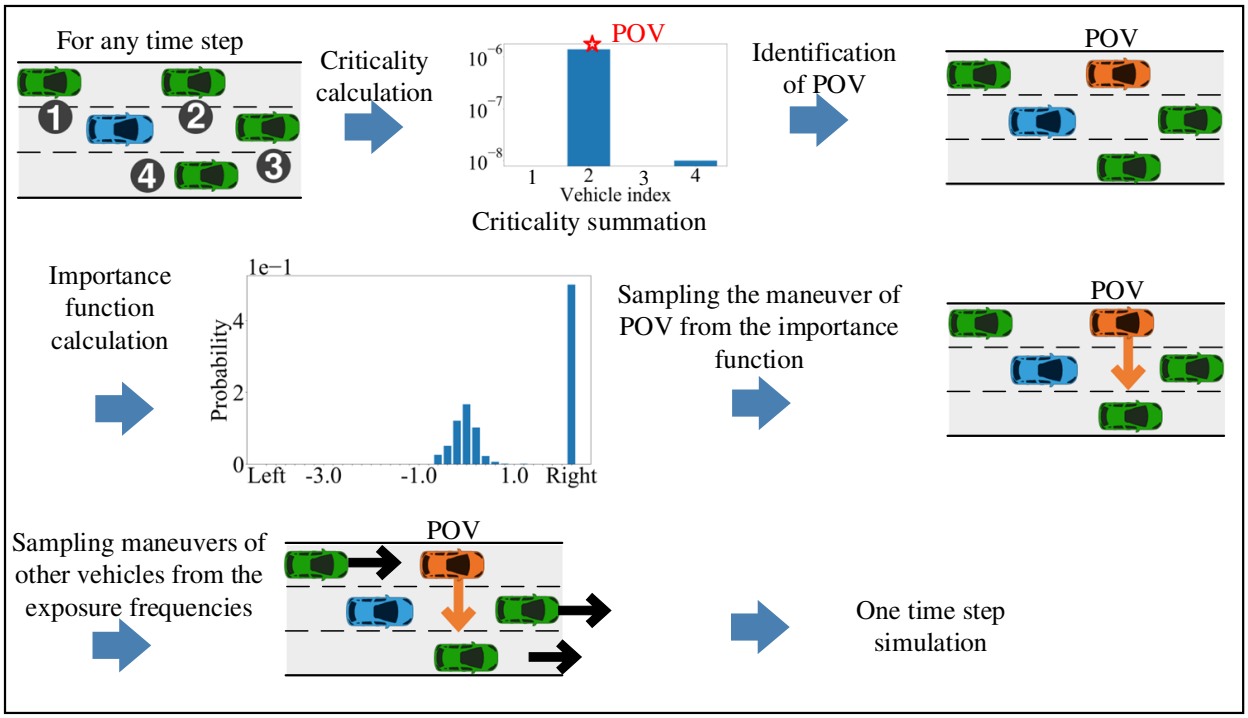


Figure 4.5: Example of NADE generation

4.3.2 Evaluation of NADE

To evaluate the generated NADE, we completed 2,000 km simulations of AVs in NDE and NADE, respectively, and calculated the distributions of bumper-to-bumper distance (range) and Time-To-Collision (TTC) for AVs. To investigate the influences of AVs, we developed two different types of AV models: the AV-I model was constructed based on IDM and MOBIL, while the AV-II model was trained by DRL techniques considering both the efficiency and safety. More details on AV-I and AV-II can be found in the Appendix C.2. Figure 4.6a shows that, for the AV-I model, NADE generates very similar distributions as NDE (naturalistic), but much more dangerous scenarios with small distances and TTC (adversarial). It is also true for the AV-II model, as shown in Figure 4.6b. The results also indicate that the AV-II model is more aggressive than the AV-I model, because the AV-II model has smaller bumper-to-bumper distances and TTC in NDE. This is not surprising because IDM and MOBIL are designed to be collision-free so AV-I is comparatively conservative.

We also compared the events encountered by the AVs in NDE and NADE. Besides the accident event, we defined the events of BV cut-in, BV hard brake, lane conflict, and AV lane change, as shown in Figure 4.7a. We queried these events with the following criteria respectively: (a) a BV cuts in AV within 1.5s Time Headway (THW); (b) a leading BV within 1.5s THW brakes harder than -3 m/s^2; (c) the AVAV and BV are within 1.5s THW longitudinally and change to the same lane simultaneously; (d) the AV changes it lane to avoid the front BV. As shown in Figure 4.7b,

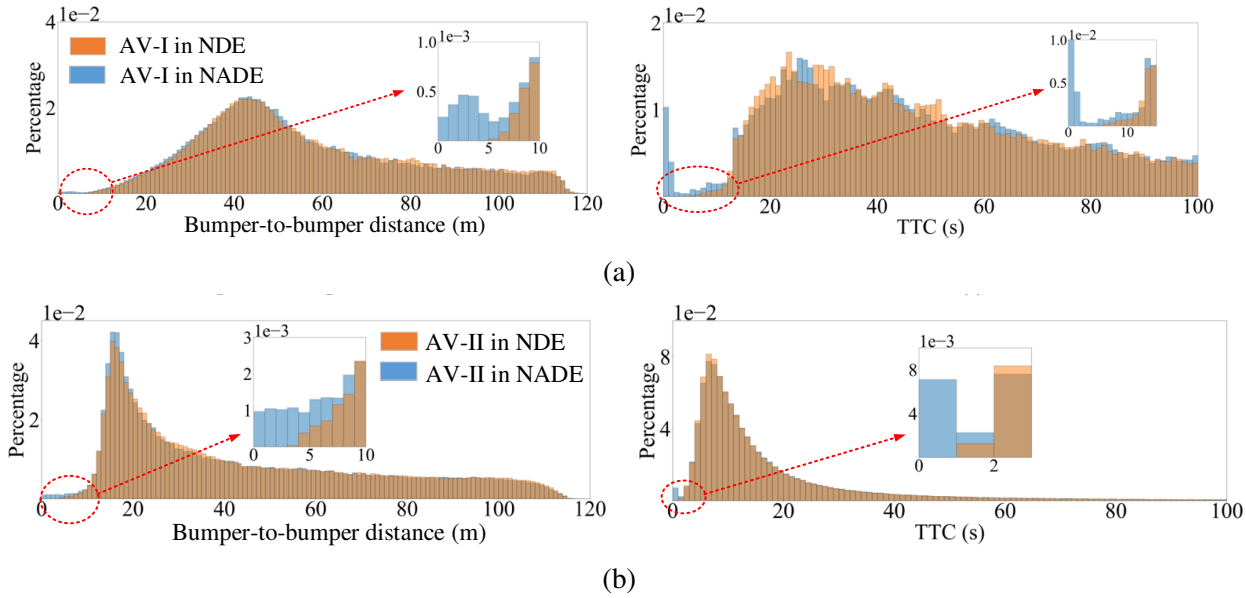


Figure 4.6: Evaluation of the generated NADE: (a) Distribution of range and TTC for the AV-I model. (b) Distribution of range and TTC for the AV-II model

comparing with NDE, NADE generates much more events of accident, BV cut-in, and lane conflict, and a similar number of BV hard brake events, for both the two AV models. Actually, NDE has no event of accident, BV cut in, and lane conflict in the 2,000 km simulations for both the AVs, because of the rareness of these events. Moreover, as shown in Figure 4.7c, NADE generates much more "evasive lane change" maneuvers of both the AVs with small relative distances (r) and speed differences (RR). All these results show that NADE can test the AVs much more effectively by more valuable events, compared with NDE.

We further investigated the adjustment frequency of BVs' maneuvers in NADE. Results show that, for every driving mile of the two AVs, we adjusted average of 6.51 and 5.43 times, respectively. As a comparison, there are a total of 381.27 and 351.01 BVs' maneuvers in the neighborhood (the closest 8 vehicles within 120m) of the AVs every mile. Therefore, we only adjust about 1.7% and 1.5% maneuvers of the environment, which is very sparse and thus keep the environment naturalistic. It validates that sparse but intelligent adjustment of NDE can significantly improve test effectiveness.

4.3.3 Accuracy and efficiency of the proposed method

The accuracy and efficiency of driving intelligence test in NADE is theoretically guaranteed and validated in our simulation. To measure the driving intelligence regarding safety, accident rates of the AVs in NDE are utilized as the benchmark. In our experiments, we compared the estimated

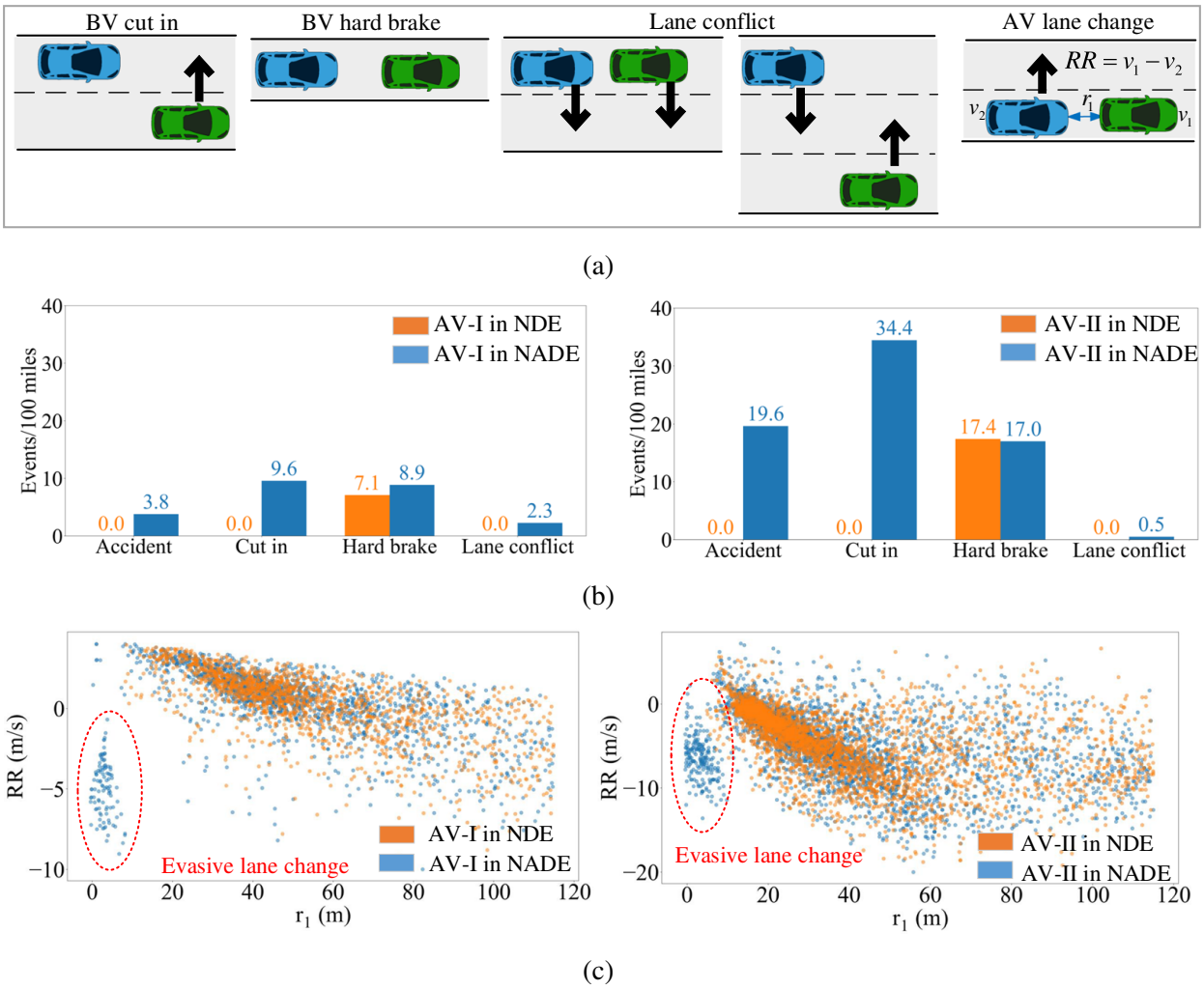


Figure 4.7: Evaluation of the generated NADE: (a) Illustration of the events. (b) The number of events encountered by the AV-I model. (c) The number of events encountered by the AV-II model

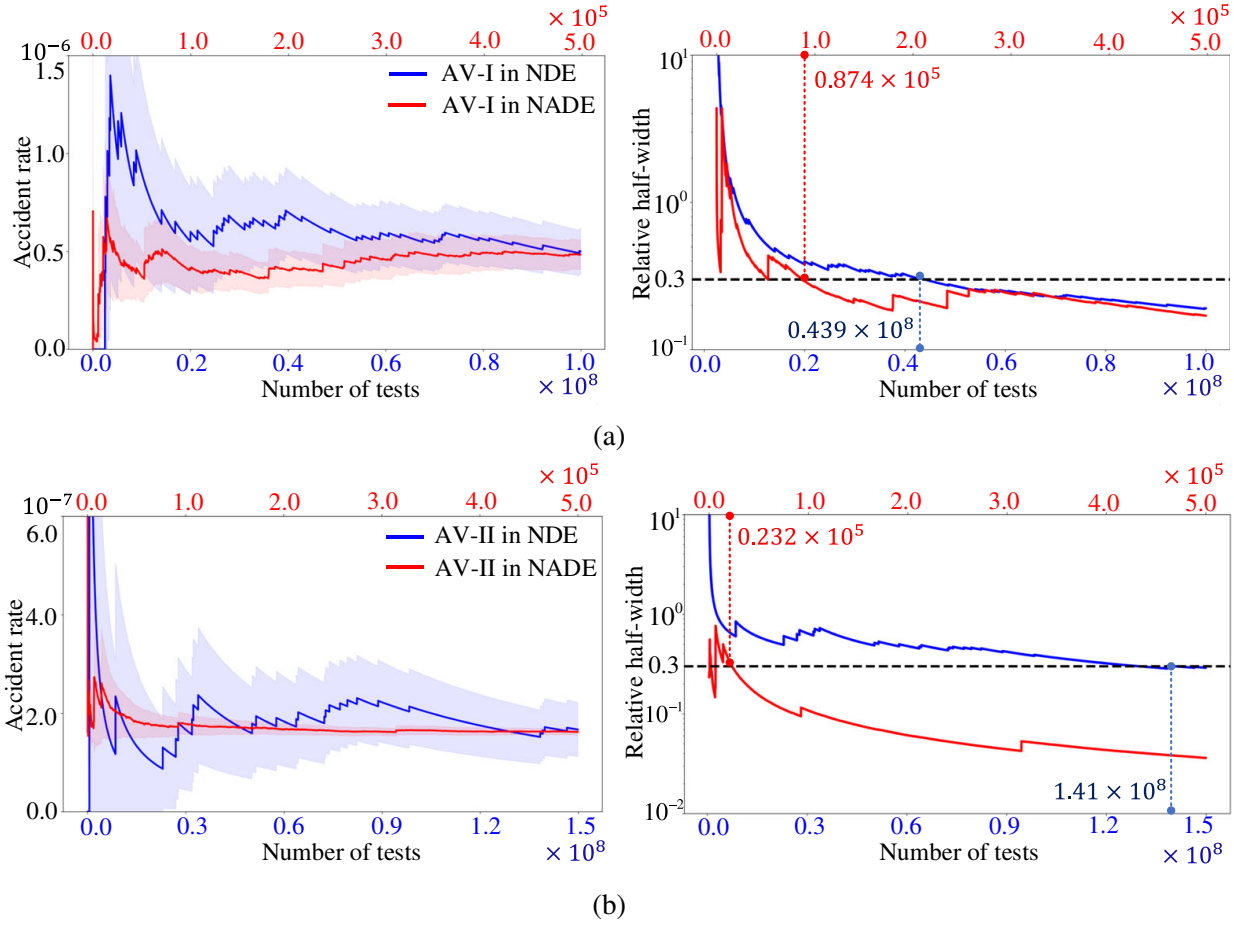


Figure 4.8: Evaluation accuracy and efficiency for the two AVs by NDE and NADE (a) AV-I model. (b) AV-II model

accident rates and required numbers of tests for both NDE and NADE. For the convenience of experiments, we conducted one simulation test for a constant driving distance (400m) of the AVs, recorded the test results (accident or not) of the AVs, and calculated the accident rate per simulation test. As the distance of each test is constant, it can easily transform between the accident rate per simulation test and the driving miles. To investigate the influences of AV models, both the AV-I and the AV-II models were tested.

Figure 4.8 shows the evaluation results of the accident rate per test for both the AVs in NDE and NADE. The blue line represents the results of testing in NDE, and the bottom x-axis indicates the number of tests. The red line represents the results in NADE, and the top y-axis for the number of tests. The light shadow represents the 90% confidence level. As shown in Figure 4.8a,c, NADE obtains the same accident rate estimation with NDE by a much smaller number of tests for both the AVs. We further calculated the average driving distance per accident, which were 5.13×10^5 and 1.54×10^6 miles per accident. As human drivers in the US have on average 4.79×10^5 miles

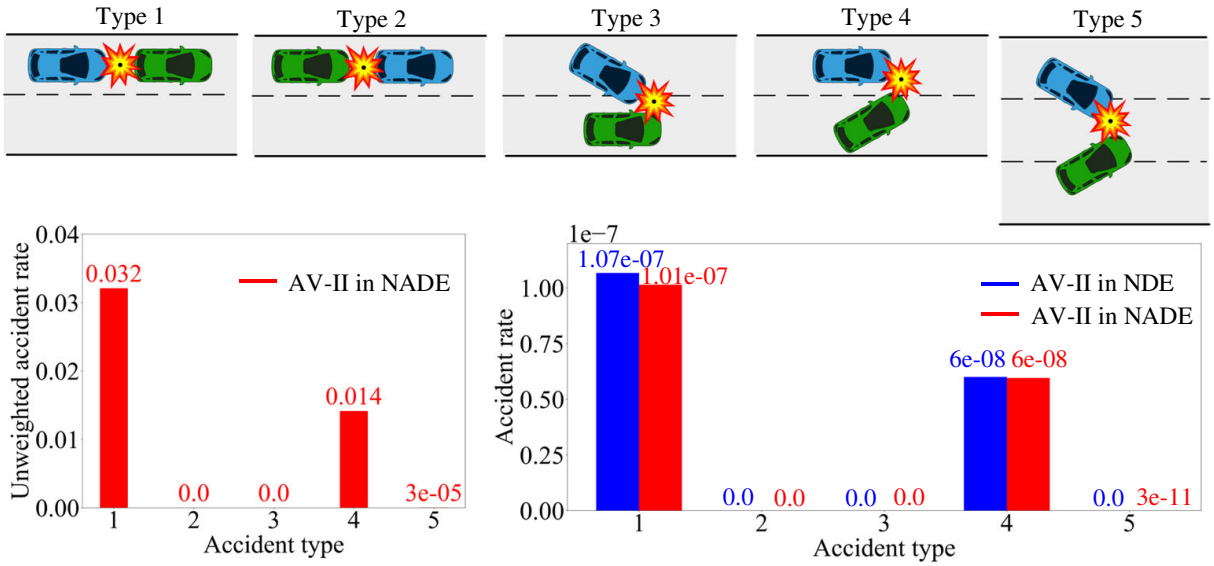


Figure 4.9: Evaluation accuracy and efficiency of accident type for AV-II

between two accidents on highway (Bureau of Transportation Statistics, 2018), the AV-I model has similar safety performance with human drivers, while the AV-II model is better.

To measure the efficiency, we calculated the Relative Half-Width (RHW) (Zhao et al., 2016) as the measurement of evaluation precision and calculated the minimum number of tests for reaching a pre-determined precision threshold (RHW is 0.3). As shown in Figure fig:AV1-crash-rate (right), for the AV-I model, NADE requires only 8.74×10^4 number of tests, while NDE requires 4.39×10^7 number of tests. Our method can accelerate the evaluation by about 500 times and reduce about 10 million driving miles. Similarly, for the AV-II model, NADE requires the 2.32×10^4 number of tests, while NDE requires 1.41×10^8 number of tests, as shown in Figure fig:AV2-crash-rate (right). Our method can accelerate the evaluation by about 6,000 times and eliminate 35 million driving miles.

To validate the unbiasedness of accident types, we adopted the crash type diagram defined by United States. Department of Transportation (2018). For the highway driving case in this chapter, we only have the accidents between the AV and BVs, so the five accident types are identified, as shown in Figure 4.9 (top). We note that the accident type 1 can also be caused by reckless cut-in of the BV, where the difference from the type 4 is that the AV collides the BV from the rear end. We have compared the results for the AV-II model in NDE and NADE. Figure 4.9 (bottom left) shows the unweighted accident rate of each type in NADE. As NADE is more adversarial than NDE, the total accident rate is 0.046 accident per simulation test, which is much larger than NDE. As required by the IS theory, each accident event should be weighted by the likelihood ratio to keep the unbiasedness. Figure 4.9 (bottom right) shows that the weighted accident rates for all five types are same as the results in NDE within the evaluation precision (the RHW is smaller than 0.3). The

Table 4.1: The average minimum number of tests and average wall-clock time in NDE and NADE

Autonomous vehicle		**NADE ($\epsilon = 0.1$)**	**NADE ($\epsilon = 0.3$)**	**NADE ($\epsilon = 0.5$)**	**NDE ($\epsilon = 1.0$)**
AV-I	No. of tests	1.85×10^5	1.52×10^5	1.14×10^5	4.39×10^7
AV-I	Time (s)	324.61	268.14	196.82	6.89×10^4
AV-II	No. of Tests	9.40×10^3	2.27×10^3	6.01×10^3	1.41×10^8
AV-II	Time (s)	17.25	4.06	10.66	2.33×10^5

summation of the accident rates of all five types is the same as the total accident rate, so these five types include all accidents of the AV-II model in both NDE and NADE.

To investigate the influences of parameters in NADE, we further conducted the sensitivity analysis of ϵ, which was used in constructing important functions. For each value (0.1, 0.3, and 0.5), we completed the tests in NADE and calculated the minimum number of tests for reaching the precision threshold. To mitigate the randomness of the results, we repeated the tests for 10 times, and calculated the average minimum number of tests, as shown in Table 4.1. Please note that NDE is equivalent to NADE with $\epsilon = 1$. Results show that NADE can improve the evaluation efficiency significantly for all three values. The best result is obtained for the AV-I model with $\epsilon = 0.5$ and AV-II model with $\epsilon = 0.3$. As discussed before, the introduction of ϵ is to mitigate the influence of approximation errors of maneuver challenges. As the approximation errors may be different for different AVs, the optimal value of ϵ and the optimal acceleration rates are different. In practice, $\epsilon = 0.5$ is a good choice balancing the optimality and the robustness.

To investigate the computational cost of NADE, we also compared the average wall-clock time used by NDE and NADE for reaching the precision threshold. We conducted the simulations of NDE and NADE on the University of Michigan's Great Lakes High-Performance Computing (HPC) cluster using 500 cores (Intel Xeon Gold 6154 processor) and 2500 GB RAM. As shown in Table 4.1, the tests in NADE reduce the computational time significantly for both AV models with all three values of ϵ.

4.3.4 Adversarial examples in NADE

We investigated the capability of NADE for generating adversarial examples. Adversarial examples have been widely investigated in the domain of Machine Learning (ML). By applying small but intentionally perturbations to examples from the dataset, adversarial examples can cause severe failures in many ML methods and, therefore, provide insights for further improvement (Goodfellow et al., 2014). Similarly, adversarial examples, sometimes known as corner cases, edge cases, or worst cases, play an important role in the development and evaluation of AVs. As they happen rarely in the NDE, it is significant to generate adversarial examples systematically. As demonstrated above, the NADE can generate many more accidents than the NDE. The key is

Figure 4.10: Adversarial examples generated in NADE

to identify the cases that are valuable and informative. We propose two criteria as examples to illustrate the potential of NADE for generating adversarial examples. The first is the simulation weight, which is the likelihood ratio of the simulation test. A smaller simulation weight usually indicates a higher probability of the test being an adversarial example. The second is the diversity of the events (as defined in Figure 4.7a) involved during the test. A test involving diverse events usually contains more information for understanding the AV model under test. Figure 4.10 provides several examples identified using the above criteria. The blue vehicle represents the AV under test, the green vehicles represent the BVs, and the green vehicle with the red rectangle represents the POV. In the first case, AV was on high speed and had a rear-end collision with the cut-in POV after two front-left POVs sequentially changed their lanes towards the AV. In the second case, the AV made a left lane change and collided with the POV due to a lane conflict (the POV accelerated first and then made a right lane change, simultaneously with the AV). In the third case, The AV turned left to avoid the collision to the cut-in POV but failed as the cut-in POV switched back to the left lane simultaneously with the AV. In the last case, the AV made an evasive lane change to avoid one cut-in POV but eventually collided with another cut-in POV.

4.4 Summary

In this chapter, we developed NADE, an intelligent testing environment based on NDE developed in Chapter 2 for AV safety performance evaluation. The proposed method can be used to enhance the high-fidelity NDE to significantly accelerate the AV testing process without loss of testing accuracy. The proposed method can also be used to systematically generate valuable adversarial examples for the further development of AVs. High-fidelity NDE is the foundation and prerequisite of NADE, and these results further demonstrate the importance of NDE for AV applications.

The efficiency of using NADE for AV testing is dependent on the approximation error of the maneuver challenge of BVs. The approximation error comes from two problems, one is the difference between the SM and the real AV under test, and the other is the prediction error of the AV maneuver in the following time steps, which is interdependent on the maneuvers of BVs. The first problem can be mitigated by prior knowledge of the AV, such as the testing results of its previous model. Although this knowledge may not be complete, it can be leveraged by our framework in constructing SM and thus reduces the difference between the SM and the AV model. The second problem is essentially a policy evaluation in the AI domain, where state-of-the-art algorithms such as those from DRL can be utilized for further reducing the approximation errors. With smaller approximation errors, the NADE can further accelerate the testing process of AVs.

For future works, ML-based methods should be developed to substitute the model-based methods and heuristics used during the POV identification. These can help extend the proposed

methodology to more complex environments, ultimately facilitating comprehensive AV evaluations at the city level for full-length trips.

CHAPTER 5

Summary and Future Directions

5.1 Summary of the dissertation

This dissertation centers on the pivotal task of simulating the high-fidelity NDE for AVs. Simulation plays an indispensable role in the development and testing of AVs, due to its unmatched advantages of controllability, repeatability, and cost-effectiveness. It serves as the cornerstone for building a closed-loop pipeline aimed at enhancing AV safety performance. However, for simulation to fulfill its promise as an effective tool, it must meet the novel and demanding criterion of statistical realism in terms of fidelity. The simulated NDE needs to be statistically representative of the real-world traffic environment, particularly for those long-tail safety-critical events, which are critical for AV safety. Existing methods, however, fall short in achieving this level of fidelity, underscoring the need for new approaches in this vital domain.

The key to NDE is modeling human driving behavior. With the advancement of sensing technologies and data acquisition systems, NDD are being collected in scale and become widely available. Therefore, in Chapter 2, we propose a data-driven method to characterize stochastic human driving behavior, harnessing the wealth of large-scale NDD. Specifically, car-following and lane-changing models in different situations are developed, in which longitudinal acceleration distributions and lane-changing probabilities are estimated using empirical distributions. To satisfy the statistical realism requirement, an optimization problem based on the Markov chain is formulated to refine empirical behavior models. The proposed method is validated in simulating a multi-lane highway driving environment using real-world data. The results show that it can reproduce accurate statistics, including speed, distance, etc., and outperforms existing methods, such as the widely used microscopic simulator SUMO.

In contrast to highway driving environments, urban environments usually involve more complex interactions between multiple agents, and the driving behavior of each agent could be difficult to model independently using car-following and lane-changing models. To address this, Chapter 3 introduces NeuralNDE - a DL-based method which characterizes the joint behavior

distribution of multiple agents. By leveraging the state-of-the-art neural network backbone (Transformer) with the supervision of NDD, it can accurately model nuanced and diverse human interactions. To enhance modeling accuracy in safety-critical situations, we develop a controllable mechanism for generating safety-critical events. This ensures the generation of safety-critical events that align with real-world occurrence frequencies and patterns. The proposed method demonstrates promising performance for simulating urban roundabout environments with statistical realism for both normal and safety-critical driving conditions. Notably, it can reproduce the accurate crash rate and even the detailed composition of crash types, severities, and near-miss statistics.

One important application of high-fidelity NDE is AV safety performance evaluation. Due to the rarity of safety-critical events in NDE, directly testing AV in NDE is exceptionally time-consuming even in simulations. To improve the testing efficiency, we propose an intelligent testing algorithm in Chapter 4. We discover that sparse but adversarial adjustments to the NDE, resulting in the NADE, can significantly reduce the required test miles without loss of evaluation unbiasedness. By training the background agents to learn when to execute what adversarial maneuver, the proposed environment becomes an intelligent environment for driving intelligence testing. We demonstrate the effectiveness of the proposed method based on the highway-driving NDE developed in Chapter 2. The results show that, compared with directly evaluating AV in NDE, the proposed NADE environment can accelerate the evaluation process by multiple orders of magnitude without loss of testing accuracy. Consequently, we provide a complete pipeline for accurate and efficient simulation-based AV testing.

In summary, this dissertation presents methodologies for building high-fidelity NDE, which is the foundation and prerequisite for simulation-based AV applications. These proposed methods pave the way for enhancing AV safety performance, which is beneficial for all stakeholders, including AV developers, customers, and regulators, and contributes to the long-term development of AV technology.

5.2 Future directions

At last, this dissertation will provide some future directions.

City-level simulation The goal for NDE is to conduct city-level or even larger-scale simulations. Currently, the proposed method (NeuralNDE) excels primarily in environments abundant with trajectory data. To enhance scalability, a pivotal avenue for future research involves the development of a unified model, which should integrate road network information into its input. By training this model on data sourced from diverse geographical locations, it should attain the

capacity to generalize its understanding to previously unseen road geometries. The substantial leap in model generalization holds the key to enabling city-level simulations, an essential stride forward in the pursuit of comprehensive acAV development and testing.

AV influence on NDE In most existing studies, including this dissertation, the assumption has been that the presence of an AV has a negligible effect on its surrounding vehicles. Nonetheless, it is crucial to recognize that human-driven vehicles might behave differently if they are interacting with an AV (Yu et al., 2021). This influence could potentially intensify as the number of AVs on the road increases. Consequently, further developments and considerations are required on NDE modeling to consider the AV influences on the NDE. Additionally, exploring the evolution and transformation process of human driving behavior in response to the growing presence of AVs on the road would be an interesting research problem.

Safety-critical behavior modeling The accuracy of human driving behavior in safety-critical situations should be further enhanced, given the significant impact minor differences can have on AV safety. Recent increases in high-resolution data collection, especially in near-miss scenarios (Ke et al., 2023), offer opportunities to develop specialized behavior models for safety-critical situations, further improving NDE fidelity.

Macroscopic-level fidelity This dissertation primarily emphasizes achieving microscopic-level fidelity in the NDE. However, it is imperative to extend this fidelity validation to the macroscopic level when simulating large road networks. This includes assessing the simulator's ability to replicate real-world traffic phenomena like traffic oscillations and the fundamental diagram. By achieving high accuracy in both microscopic and macroscopic statistics, we anticipate the NDE simulator will assume a pivotal role in diverse domains encompassing AVs and the development of intelligent transportation systems.

APPENDIX A

Additional Details in Chapter 2

A.1 Naturalistic driving data processing

First, the original data were segmented into trajectories with the following requirements: (1) the time should be continuous and does not have discontinuity of more than 2 seconds, (2) the identification of the surrounding vehicles is consistent throughout the trajectory, (3) each trajectory should last more than 3 seconds. Data that are too noisy (with severe speed and/or position oscillations) are discarded.

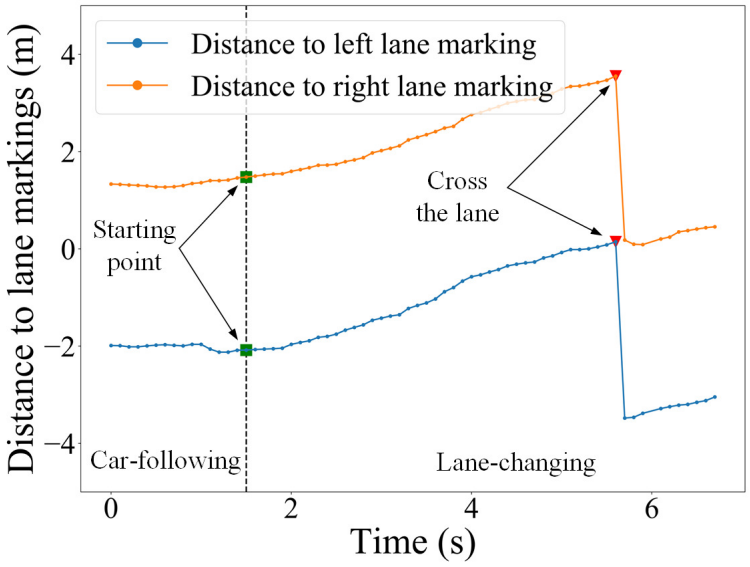


Figure A.1: Lane-changing events identification example: the subject vehicle is doing a left lane change.

Second, considering the driving environment of the subject vehicle, trajectory data points will be categorized into the six situations as illustrated in Figure 2.3a, correspondingly. An important step in the categorization is to identify the lane-changing event by analyzing the lateral distance to the lane markings. Figure A.1 shows an example that the ego-vehicle is doing a left lane change.

Note that the sign of the distance to lane marking differentiates the left and right lane markings. Figure A.1 shows the whole process of a left lane-changing event: the vehicle is approaching the left lane marking and crosses the lane when the distance to the left lane marking equals zero, and then changes to the maximum (lane width) after entering the target lane. The starting moment and crossing the lane moment of lane-changing events can be identified based on both the distance and slope change of the distance, accordingly. Similar techniques are also utilized for lane-changing detection in a recent study (De Gelder et al., 2020). The starting point of the lane-changing event will be used for calculating the lane-change probability.

A.2 Experiment settings

The experiment settings of the proposed method and the SUMO baseline are discussed in the section. For the proposed method, a three-lane highway driving environment is developed based on an open-source highway traffic simulator (Leurent, 2018). The bicycle model is implemented to update vehicle states at a 10Hz frequency. All lane-changing maneuvers are set completed in 1 second. To be consistent with the NDD, the speed of the NDE simulation is bounded between 20 to 40 m/s, and the longitudinal acceleration is bounded between -4 to 2 m/s^2. Since both the longitudinal and lateral models are data-driven, there are inevitably states where no NDD is collected. We will use the IDM and MOBIL models for states not covered by the proposed behavior models. The IDM model parameters are calibrated from the same NDD. The MOBIL model parameters are partially from (Gong et al., 2018) which are also calibrated using the SPMD dataset. The detailed model parameters are listed in the next paragraph. The simulation initialization method determines the initial state of all vehicles, which includes the position and speed, etc. A realistic initialization is preferred to shorten the required warm-up time, which can improve the simulation efficiency. In this study, a data-driven initialization method is proposed based on NDD and the details can be found in Appendix A.3. Using the proposed initialization method, around 60 vehicles will be initialized for each simulation. It is approximately 1360 vehicles/hour/lane, which belongs to level of service (LOS) C for multilane highways (Federal Highway Administration (FHWA), 1994). We ran 100 simulations to mitigate the randomness effect. To fully examine the error accumulation issue, each simulation ran 15 minutes, which included 10 minutes of warm-up time and 5 minutes of data collection.

The calibrated parameters of the IDM model using the SPMD are: maximum acceleration (0.8 m/s^2), desired speed (37 m/s), exponent parameter (3.0), comfortable deceleration (-1.3 m/s^2), gap at standstill (0.1 m), and desired time headway (0.8 s). To generate a stochastic version IDM, we added zero-mean Gaussian noise to the standard IDM output based on existing literature (Treiber and Kesting, 2017; Laval et al., 2014). We examined different standard deviation

parameters from 0.5 to 3.0 with a resolution of 0.5. We presented the results with the best performance (standard deviation equals 1.5). The parameters of the MOBIL model are: politeness factor (0.1), utility threshold (0.2 m/s^2), and maximum safe deceleration (-3 m/s^2). The calibrated parameters of the W99 model are: minGap (2.0 m), CC1 (1.2 s), CC2 (9.9 m), CC3 (-19.3 s), CC4 (-0.2 m/s), CC5 (0.1 m/s), CC6 (1.5 $10^{-4} rad/s$), CC7 (1.0 m/s^2), CC8 (0.6 m/s^2), CC9 (0.5m/s^2). For the SUMO baseline, all simulation settings are the same as the proposed method. Each simulation lasts for 15 minutes, which includes 10 minutes of warm-up time and 5 minutes of data collection. We set the input traffic flow as 1360 vehicles/hour/lane and 60 vehicles will be generated in each simulation. The initial speed of the vehicle is set as 32 m/s based on NDD. To account for the stochasticity of the simulator, we ran 100 simulations for each model to collect data.

A.3 The proposed data-driven simulation initialization method

We propose a data-driven initialization method to sequentially determine the $(i+1)$-th vehicle state (x_i, y_i, v_i), i.e., longitudinal position, lateral position, and velocity, of downstream vehicles based on its upstream i-th vehicles' states. The lateral position y_i is the same as the lateral coordinates of its lane center. The first vehicle of each lane is determined by sampling its longitudinal position inside an initial zone from a uniform distribution and its velocity from the empirical velocity distribution. This can be expressed as

$$x_i \sim U\left(0, d_0\right), \tag{A.1}$$

$$v_i \sim \pi_v^{\star}, \tag{A.2}$$

where $U\left(\cdot,\cdot\right)$ denotes the uniform distribution, d_0 is the predetermined initial zone size, v_0 is the speed, and $\pi_v^{\star}$ is the empirical velocity distribution obtained from the NDD. Based on the state of the i-th vehicle, we can determine its downstream $(i+1)$-th vehicle state. An indicator variable follows a Bernoulli distribution $I_i \sim B\left(p_{CF}\right)$ is sampled to determine whether the i-th vehicle is in the car-following $(I_i = 1)$ situation or free-driving $(I_i = 0)$ situation. p_{CF} is the probability a vehicle is in the car-following case, which can be approximated from NDD. Then the joint distribution of range r and range rate rr given the current speed $g(r, rr|v_i)$ is queried from the NDD if $I_i = 1$. Then, the position and velocity of the next (downstream) vehicle can be obtained by

$$x_{i+1} = x_i + r, \tag{A.3}$$

$$y_{i+1} = y_i, \tag{A.4}$$

$$v_{i+1} = v_i + rr, \tag{A.5}$$

where r, rr is the range and range rate sampled from $g(r, rr|v_i)$, and the lateral position of the $(i+1)$-th vehicle is the same as the i-th vehicle since they are in the same lane. If $I_i = 0$, the next vehicle is randomly generated in a new initial zone outside the car-following observation range, and the velocity follows the empirical velocity distribution. It can be represented as

$$x_{i+1} = x_i + d_{obs} + r, \tag{A.6}$$

$$r \sim U\left(0, d_0\right), \tag{A.7}$$

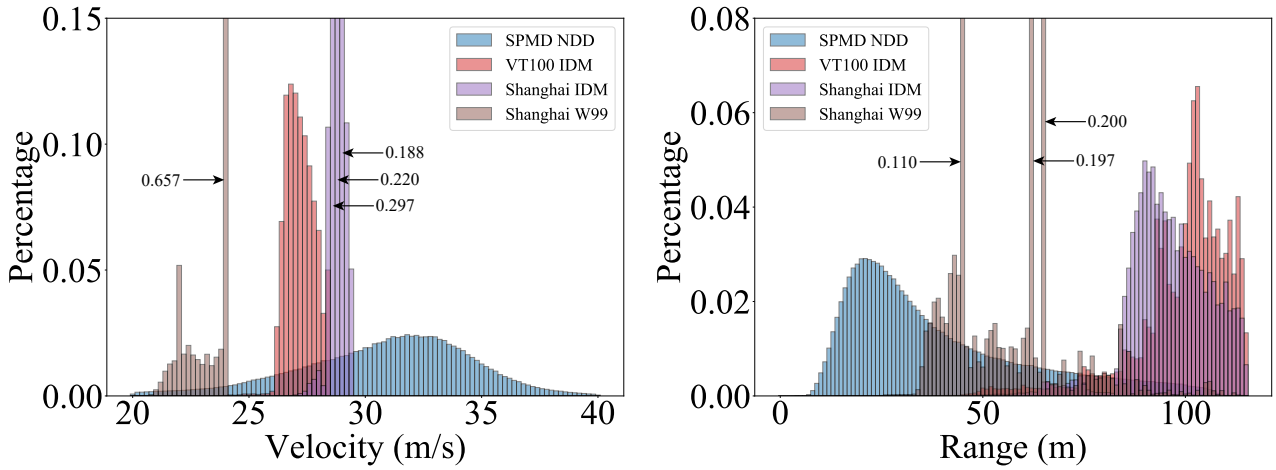
$$y_{i+1} = y_i, \tag{A.8}$$

$$v_{i+1} \sim \pi_v^\star, \tag{A.9}$$

where d_{obs} is the predetermined car-following observation range suggested by the NDD. Vehicles on each lane can be sequentially generated by repeating this process. Note that unrealistic initialization that leads to trivial initial collision will be rejected and resampled. Based on the data, the parameters used in the study are $d_0 = 50m$, $p_{CF} = 0.68$, and $d_{obs} = 115m$.

A.4 Performance of baseline methods using parameters from the literature

Besides using parameters calibrated by ourselves, we also use model parameters from the literature to fully illustrate the performance of existing methods which includes those calibrated by the NDD from Virginia, US (Sangster et al., 2013) (denoted as VT100 IDM), and Shanghai, China Zhu et al. (2018a) (denoted as Shanghai IDM and Shanghai W99). The vehicle initial speed is set based on the parameters of each model, which is 26 m/s for VT100 IDM, 28 m/s for Shanghai IDM, and 22 m/s for Shanghai W99. Other simulation settings are the same as discussed in Appendix A.2., where each model runs 20 simulations to collect data. The results are shown in Figure A.2. Although the NDD reported in Figure A.2 is not the same one used for calibration in the literature, the comparison is still informative. The distributions generated using existing methods are concentrated in relatively dense regions, however, the real-world distribution spreads naturally



(a) (b)

Figure A.2: Velocity and range distributions that generated using model parameters from the literature.

among the whole intervals.

APPENDIX B

Additional Details in Chapter 3

B.1 NeuralNDE results of rounD dataset

The NeuralNDE results using the rounD dataset are shown in Figure B.1. From the results, we can find that NeuralNDE can achieve statistical realism and significantly outperform existing methods. We only show normal driving behavior statistics since the safety-critical driving behavior ground truth, e.g., crash and near-miss data, is unavailable.

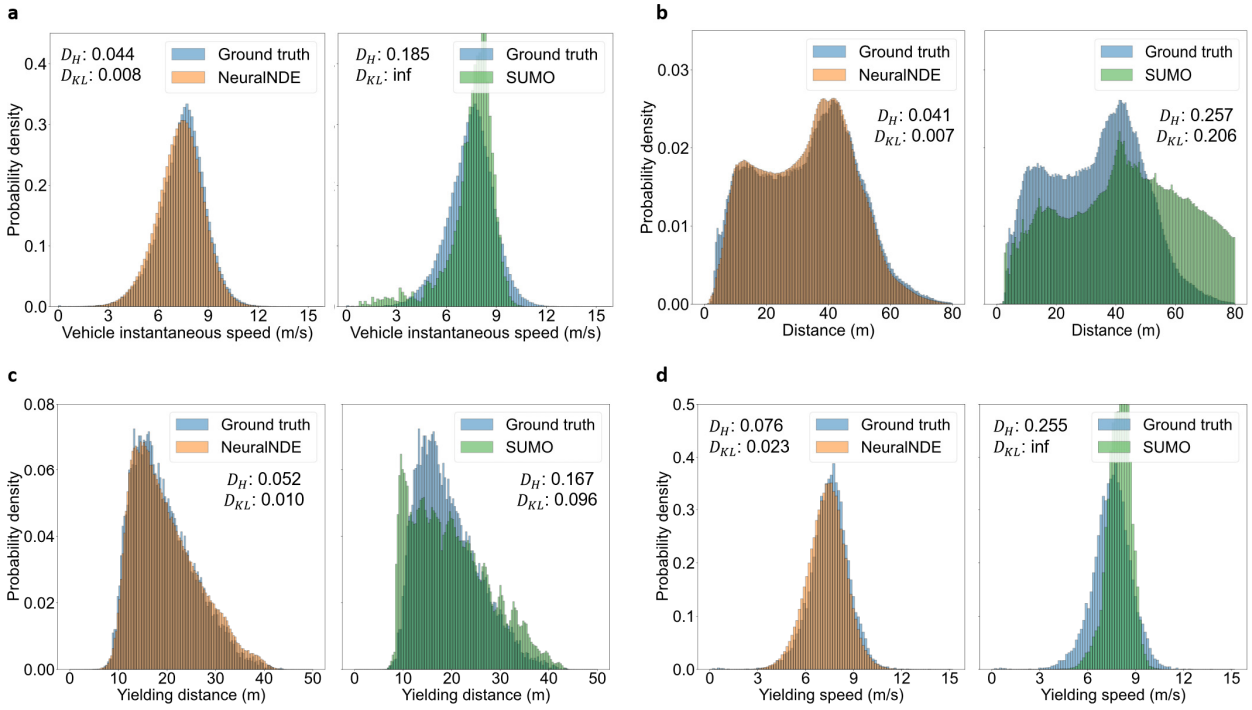


Figure B.1: Statistical realism of NeuralNDE using rounD dataset. (a) Vehicle instantaneous speed distribution. (b) Vehicle distance distribution. (c) Yielding distance distribution. (d) Yielding speed distribution

B.2 SUMO simulation settings

The IDM (Treiber et al., 2000) and SL2015 model are used as the car-following model and lane-changing model, respectively. The SUMO default SL2015 lane-changing model is used with default parameters. The IDM model considers both the impact of the desired speed and the desired bumper-to-bumper range on the longitudinal behavior, as shown below

$$a\left(t\right) = a_{max}\left[1 - \left(\frac{v(t)}{v_0}\right)^{\delta} - \left(\frac{s^{\star}(t)}{s(t)}\right)^{2}\right], \tag{B.1}$$

where a_{max} is the maximum acceleration, v_0 is the desired speed, $v(t)$ and $s(t)$ are the velocity and the bumper-to-bumper range at the current time step t, δ is an exponent parameter and $s^{\star}$ is the desired bumper-to-bumper range which can be calculated as

$$s^{\star} = s_0 + \max\left(0, v(t)\cdot T - \frac{v(t)\cdot\Delta v(t)}{2\sqrt{a_{max}\cdot a_{comf}}}\right), \tag{B.2}$$

where s_0 is the minimum range at a standstill, T is the desired time headway, $\Delta v(t)$ is the speed difference and a_{comf} is the comfortable deceleration. The model parameters are set based on the data and common practice as follows: $v_0 = 12.5m/s, T = 1.0s, s_0 = 2m, a_{max} = 9.0m/s^2, a_{comf} = 4.5m/s^2, \delta = 4$. Note that different sets of parameters are examined and results with the best performance are used and reported in the paper.

B.3 Crash type and severity details

To determine the crash type of a simulated collision, we follow the definition of the National Highway Traffic Safety Administration (2022) and consider the state of collision vehicles at the crash moment. Specifically, we consider the relative position and relative heading of the two colliding vehicles. There are four potential relative positions, i.e., front, left, right, and rear, of two colliding vehicles, as shown in Figure B.2. The relative heading of two vehicles is between 0 to 180 degrees, where 0 degrees means two vehicles are heading the same direction and 180 degrees means the opposite direction. We define a crash as rear-end if the relative position is rear or front, and the relative heading is smaller than 40 degrees. A sideswipe crash is when the relative position is left or right, and the relative heading is smaller than 30 degrees or greater than 150 degrees. A head-on crash is when the relative position is front, and the relative heading is greater than 90 degrees. Other crashes are considered angle crashes.

To calculate the Delta-V based on the conservation of momentum, the collision is assumed to be a perfectly inelastic collision and the vehicles have the same mass. Therefore, the change

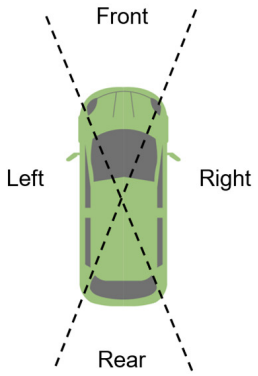


Figure B.2: Vehicle relative position for crash type classification

in velocity can be obtained based on the difference between the impact speed vector and the separation speed vector. For example, consider a rear-end crash, where the front vehicle is initially stationary and the rear-end collision vehicle is traveling at 30 mph. Then, the separation speed of the two vehicles will be 15 mph, and the Delta-V of both vehicles are 15 mph. Many existing studies investigated the relationship between Delta-V and occupant injury level, we follow their found thresholds (Richards, 2010) to measure the crash severity. Specifically, in side impact crashes (e.g., angle crash), there is no injury if Delta-V is smaller than 8 mph, minor injury if Delta-V is between 8 and 14 mph, serious injury if Delta-V is between 14 and 24 mph, and fatal injury if Delta-V is greater than 24 mph. For frontal impact crashes (e.g., rear-end crash), the corresponding thresholds are no injury $(0, 11]$ mph, minor injury $(11, 23]$, serious injury $(23, 34]$, and fatal injury $(34, \infty)$.

B.4 Model scalability experiment settings

We use SUMO (Lopez et al., 2018) to generate vehicle trajectory data and use it as the ground truth of NDE for training and validation. We collect around five hours of data to train NeuralNDE models for the intersection and roundabout scenarios. The training settings are the same as previous experiments using AA and rounD datasets. During the inference time, the intersection and roundabout areas are controlled by NeuralNDE, and the transition area in between is controlled by model-based methods. Since there is no crash in NDE (i.e., SUMO), the acceptance probability of the crash critic module is set to zero. We also apply the safety mapping rule that is used to train the safety mapping network to the whole simulation environment including the transition area to further guarantee the safety of the simulation. We simulate the whole network for around 100 hours and collect the data to validate the performance. The simulation resolutions and metrics definitions are the same as in previous experiments. The PET is collected for vehicles within the roundabout

circle and intersection. The instantaneous speed for the intersection scenario is collected for all vehicles in the area.

APPENDIX C

Additional Details in Chapter 4

C.1 Construction of surrogate models

This section describes the details of the SMs used in this paper. Specifically, we constructed the SMs based on IDM car-following model (Treiber et al., 2000) and MOBIL lane-changing model (Kesting et al., 2007). The details of IDM model can be found in Appendix B.2.

The MOBIL model calculates the utility of a lane-changing maneuver as

$$u = \tilde{a}_{ego} - a_{ego} + p \cdot ((\tilde{a}_{new} - a_{new}) + (\tilde{a}_{old} - a_{old})), \tag{C.1}$$

where p is the politeness factor (0.1), $a_{ego}, a_{new}, a_{old}$ denote the original accelerations of the ego vehicle, the new follower in the target lane, and the old follower in the current lane, and $\tilde{a}_{ego}, \tilde{a}_{new}, \tilde{a}_{old}$ denote the new accelerations if the ego vehicle changes its lane. To capture the randomness of AVs, we modified the model as a probabilistic lane-changing model. First, the total lane change probability (p_{LC}) is calculated by using a piecewise linear function of the total utility $(u_T = \max\{0, u_L\} + + \max\{0, u_R\})$:

$$p_{LC} = \begin{cases} 0.9, & u_T \geq 1 \\ (0.9 - 2 \times 10^{-8})u_T, & u_T \in (0, 1) \\ 2 \times 10^{-8}, & u_T = 0 \end{cases} \tag{C.2}$$

where 2×10^{-8} indicates the minimum probability of lane change. Then, the specific probability for the left and right lane-changing maneuver is proportionally allocated according to its utility u_L and u_R.

C.2 Construction of AV models

This section describes the details of the two AV models used in Chapter 4. The AV-I model was constructed by the IDM and MOBIL models with the calibrated parameters: $a_{max} = 1.5m/s^2, v_0 = 33.33m/s, \delta = 4, s_0 = 2.0m, T = 1.2s, a_{comf} = 2m/s^2, p = 0.0$. The AV-II model was trained by DRL techniques with the efficiency reward mapping the velocity from $[20, 40]m/s$ to the reward $[-1, 1]$. The inputs of the network were the states (velocity, position, and lane ID) of the vehicles within the AV's observation range (120 m). The outputs were the 33-dimensional discretized maneuver space, including left and right lane changes, and the 31 discrete longitudinal accelerations from $[-4, 2]m/s^2$ with a resolution of $0.2m/s^2$. The AV agent predicted behaviors of surrounding vehicles for one time step and avoided the maneuvers that lead to immediate accidents. We implemented the double deep Q-networks (DDQN) algorithm (Van Hasselt et al., 2016) with dueling networks (Wang et al., 2016) to train the AV agent. Besides the dueling layer, we used two hidden layers with 256 neurons. We set the learning rate (10^{-5}), batch size (64), discount factor (0.99), target network update frequency (1,000), replay memory size (50,000), loss function (mean square error), and the optimizer (Adam), as commonly used in the practices.

C.3 Proof of Theorem 4.1

By the IS theory, the estimation variance can be derived as

$$\sigma^2 = E_{q(x)}\left(P^2\left(A|x\right)\frac{P^2(x)}{q^2(x)}\right) - P^2(A), \tag{C.3}$$

where $q(x) = P(x_{-c})q(x_c)$ denotes that the critical variables are sampled from the importance function $x_c \sim q(x_c)$, while other variables are sampled from the naturalistic distributions $x_{-c} \sim P(x_{-c})$. Substituting $q(x)$ into the equation, we have the following equation as

$$\begin{aligned}\sigma^2 &= E_{q(x)}\left(P^2\left(A|x\right)\frac{P^2(x_c)}{q^2(x_c)}\right) - P^2(A), \\ &= E_{q(x_c)}\left(E_{q(x_{-c})}\left(P^2\left(A|x\right)\right)\frac{P^2(x_c)}{q^2(x_c)}\right) - P^2(A).\end{aligned} \tag{C.4}$$

Substituting the variance equation that

$$E_{P(x_{-c})}\left(P^2\left(A|x\right)\right) = E^2_{P(x_{-c})}\left(P(A|x)\right) + \sigma^2_{P(x_{-c})}(P(A|x)), \tag{C.5}$$

the estimation variance equation can be further derived as

$$\begin{aligned}\sigma^2 &= E_{q(x)}\left(P^2\left(A|x\right)\frac{P^2(x_c)}{q^2(x_c)}\right) - P^2(A),\\ &= E_{q(x_c)}\left(E^2_{P(x_{-c})}\left(P(A|x)\right)\frac{P^2(x_c)}{q^2(x_c))}\right) + E_{q(x_c)}\left(\sigma^2_{P(x_{-c})}(P(A|x))\frac{P^2(x_c)}{q^2(x_c))}\right) - P^2(A), \quad \text{(C.6)}\end{aligned}$$

The first term can be derived as

$$E_{q(x_c)}\left(E^2_{P(x_{-c})}\left(P(A|x)\right)\frac{P^2(x_c)}{q^2(x_c))}\right) = E_{q(x_c)}\left(P^2(A|x_c)\frac{P^2(x_c)}{q^2(x_c)}\right). \quad \text{(C.7)}$$

By the IS theory, the optimal importance function for the critical variables is

$$q^\star(x_c) = \frac{P(A|x_c)P(x_c)}{P(A)}. \quad \text{(C.8)}$$

Substituting it into the equation, the first term can be further derived as

$$\begin{aligned}E_{q(x_c)}\left(P^2(A|x_c)\frac{P^2(x_c)}{q^2(x_c)}\right) &= P^2(A)E_{q^\star(x_c)}\left(\frac{q^\star(x_c)}{q(x_c)}\right),\\ &= P^2(A)E_{q^\star(x_c)}\left(\exp\log\frac{q^\star(x_c)}{q(x_c)}\right),\\ &\geq P^2(A)\exp E_{q^\star(x_c)}\left(\log\frac{q^\star(x_c)}{q(x_c)}\right),\\ &= P^2(A)\left\{\exp\left[D_{KL}(q^\star(x_c)\|q(x_c))\right]-1\right\}. \quad \text{(C.9)}\end{aligned}$$

Moreover, according to the definition of χ^2-divergence, we can also derive the equivalence as

$$\begin{aligned}E_{q^\star(x_c)}\left(\frac{q^\star(x_c)}{q(x_c)}\right) &= E_{q(x_c)}\left(\left(\frac{q^\star(x_c)}{q(x_c)}\right)^2\right)+1,\\ &= D_{\chi^2}\left(q^\star(x_c)\|q(x_c)\right)+1. \quad \text{(C.10)}\end{aligned}$$

The second term can be derived as

$$E_{q(x_c)}\left(\sigma^2_{P(x_{-c})}(P(A|x))\frac{P^2(x_c)}{q^2(x_c))}\right) = E_{q(x)}\left[\left(P(A|x)-P(A|x_c)\right)^2\frac{P^2(x)}{q^2(x)}\right]. \quad \text{(C.11)}$$

Substituting both the first and second terms, we can conclude the theorem.

BIBLIOGRAPHY

Andrews, J. (2023). How autonomous vehicles are making transportation more inclusive. https://media.maymobility.com/May-Mobility-Cities-Today-Whitepaper-2023.pdf.

Au, S.-K. and Beck, J. (2003). Important sampling in high dimensions. *Structural safety*, 25(2):139–163.

Bando, M., Hasebe, K., Nakayama, A., Shibata, A., and Sugiyama, Y. (1995). Dynamical model of traffic congestion and numerical simulation. *Physical review E*, 51(2):1035.

Bergamini, L., Ye, Y., Scheel, O., Chen, L., Hu, C., Del Pero, L., Osiński, B., Grimmett, H., and Ondruska, P. (2021). Simnet: Learning reactive self-driving simulations from real-world observations. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5119–5125. IEEE.

Bezzina, D. and Sayer, J. (2014). Safety pilot model deployment: Test conductor team report. *Report No. DOT HS*, 812(171):18.

Bhattacharyya, R. P., Phillips, D. J., Wulfe, B., Morton, J., Kuefler, A., and Kochenderfer, M. J. (2018). Multi-agent imitation learning for driving simulation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1534–1539. IEEE.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Bureau of Transportation Statistics (2018). Transportation accidents by mode. https://www.bts.gov/content/transportation-accidents-mode.

California Department of Motor Vehicles (2023). 2022 autonomous vehicle disengagement reports. https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/disengagement-reports/.

Chai, Y., Sapp, B., Bansal, M., and Anguelov, D. (2019). Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*.

Chen, X., Li, L., and Zhang, Y. (2010). A markov model for headway/spacing distribution of road traffic. *IEEE Transactions on Intelligent Transportation Systems*, 11(4):773–785.

Cruise LLC (2020). Cruise autonomous vehicles drive 200,000 hours a day in simulation. https://www.youtube.com/watch?v=9G2zUUTeG1I&ab_channel=Cruise.

De Gelder, E., Manders, J., Grappiolo, C., Paardekooper, J.-P., Den Camp, O. O., and De Schutter, B. (2020). Real-world scenario mining for the assessment of automated vehicles. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR.

Dowling, R., Skabardonis, A., Alexiadis, V., et al. (2004). Traffic analysis toolbox, volume iii: Guidelines for applying traffic microsimulation modeling software. Technical report, United States. Federal Highway Administration. Office of Operations.

Erdmann, J. (2015). SUMO's lane-changing model. In *Modeling Mobility with Open Data*, pages 105–123. Springer.

Federal Highway Administration (FHWA) (1994). *Highway Capacity Manual, Special Report 209*. Transportation Research Board, National Research Council.

Feng, S., Feng, Y., Sun, H., Bao, S., Zhang, Y., and Liu, H. X. (2020a). Testing scenario library generation for connected and automated vehicles, part II: Case studies. *IEEE Transactions on Intelligent Transportation Systems*, 22(9):5635–5647.

Feng, S., Feng, Y., Sun, H., Zhang, Y., and Liu, H. X. (2020b). Testing scenario library generation for connected and automated vehicles: An adaptive framework. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):1213–1222.

Feng, S., Feng, Y., Yan, X., Shen, S., Xu, S., and Liu, H. X. (2020c). Safety assessment of highly automated driving systems in test tracks: A new framework. *Accident Analysis & Prevention*, 144:105664.

Feng, S., Feng, Y., Yu, C., Zhang, Y., and Liu, H. X. (2020d). Testing scenario library generation for connected and automated vehicles, part I: Methodology. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1573–1582.

Feng, S., Sun, H., Yan, X., Zhu, H., Zou, Z., Shen, S., and Liu, H. X. (2023). Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615(7953):620–627.

Feng, S., Yan, X., Sun, H., Feng, Y., and Liu, H. X. (2021). Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature communications*, 12(1):1–14.

Gao, J., Sun, C., Zhao, H., Shen, Y., Anguelov, D., Li, C., and Schmid, C. (2020). VectorNet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533.

Gong, X., Guo, Y., Feng, Y., Sun, J., and Zhao, D. (2018). Evaluation of the energy efficiency in a mixed traffic with automated vehicles and human controlled vehicles. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1981–1986. IEEE.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Grimmett, G. and Stirzaker, D. (2020). *Probability and random processes*. Oxford university press.

Gu, J., Sun, C., and Zhao, H. (2021). DenseTNT: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312.

Guo, Q., Li, L., and Ban, X. J. (2019). Urban traffic signal control with connected and automated vehicles: A survey. *Transportation research part C: emerging technologies*, 101:313–334.

Gurobi Optimization, LLC (2022). Gurobi Optimizer Reference Manual.

Gursten, S. M. (2021). What Michigan roundabouts were the most dangerous in 2020. https://www.michiganautolaw.com/blog/2021/08/18/michigan-roundabouts-most-dangerous-2020.

Hamdar, S. H., Mahmassani, H. S., and Treiber, M. (2015). From behavioral psychology to acceleration modeling: Calibration, validation, and exploration of drivers' cognitive and safety parameters in a risk-taking environment. *Transportation Research Part B: Methodological*, 78:32–53.

Hammit, B. (2018). A case for online traffic simulation: Systematic procedure to calibrate car-following models using vehicle data. https://github.com/bhammit1/Analysis.

Hammit, B., James, R., and Ahmed, M. (2018). A case for online traffic simulation: Systematic procedure to calibrate car-following models using vehicle data. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3785–3790. IEEE.

Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2.

Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. *Advances in neural information processing systems*, 29.

Huang, Z., Liu, H., Wu, J., and Lv, C. (2023). Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving. *IEEE transactions on neural networks and learning systems*.

Igl, M., Kim, D., Kuefler, A., Mougin, P., Shah, P., Shiarlis, K., Anguelov, D., Palatucci, M., White, B., and Whiteson, S. (2022). Symphony: Learning realistic and diverse agents for autonomous driving simulation. *arXiv preprint arXiv:2205.03195*.

Jacobstein, N. (2019). Autonomous vehicles: An imperfect path to saving millions of lives.

James, S., Wohlhart, P., Kalakrishnan, M., Kalashnikov, D., Irpan, A., Ibarz, J., Levine, S., Hadsell, R., and Bousmalis, K. (2019). Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12627–12637.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

Kalra, N. and Paddock, S. M. (2016). Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193.

Kamenev, A., Wang, L., Bohan, O. B., Kulkarni, I., Kartal, B., Molchanov, A., Birchfield, S., Nistér, D., and Smolyanskiy, N. (2021). Predictionnet: Real-time joint probabilistic traffic prediction for planning, control, and simulation. *arXiv preprint arXiv:2109.11094*.

Ke, R., Cui, Z., Chen, Y., Zhu, M., Yang, H., Zhuang, Y., and Wang, Y. (2023). Lightweight edge intelligence empowered near-crash detection towards real-time vehicle event logging. *IEEE Transactions on Intelligent Vehicles*.

Kesting, A., Treiber, M., and Helbing, D. (2007). General lane-changing model MOBIL for car-following models. *Transportation Research Record*, 1999(1):86–94.

Kopelias, P., Demiridi, E., Vogiatzis, K., Skabardonis, A., and Zafiropoulou, V. (2020). Connected & autonomous vehicles–environmental impacts–a review. *Science of the total environment*, 712:135237.

Krajewski, R., Moers, T., Bock, J., Vater, L., and Eckstein, L. (2020). The round dataset: A drone dataset of road user trajectories at roundabouts in germany. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE.

Kuefler, A., Morton, J., Wheeler, T., and Kochenderfer, M. (2017). Imitating driver behavior with generative adversarial networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 204–211. IEEE.

Laval, J. A., Toth, C. S., and Zhou, Y. (2014). A parsimonious model for the formation of oscillations in car-following models. *Transportation Research Part B: Methodological*, 70:228–238.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Lee, S. E., Olsen, E. C., Wierwille, W. W., et al. (2004). A comprehensive examination of naturalistic lane-changes. Technical report, United States. Department of Transportation. National Highway Traffic Safety Administration.

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

Li, L. and Chen, X. M. (2017). Vehicle headway modeling and its inferences in macroscopic/microscopic traffic flow theory: A survey. *Transportation Research Part C: Emerging Technologies*, 76:170–188.

Li, L., Wang, X., Wang, K., Lin, Y., Xin, J., Chen, L., Xu, L., Tian, B., Ai, Y., Wang, J., et al. (2019a). Parallel testing of vehicle intelligence via virtual-real interaction. *Science robotics*, 4(28):eaaw4106.

Li, N., Oyler, D. W., Zhang, M., Yildiz, Y., Kolmanovsky, I., and Girard, A. R. (2017). Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems. *IEEE Transactions on control systems technology*, 26(5):1782–1797.

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

Li, W., Pan, C., Zhang, R., Ren, J., Ma, Y., Fang, J., Yan, F., Geng, Q., Huang, X., Gong, H., et al. (2019b). AADS: Augmented autonomous driving simulation using data-driven algorithms. *Science robotics*, 4(28):eaaw0863.

Liu, H. X. and Feng, S. (2022). "curse of rarity" for autonomous vehicles. *arXiv preprint arXiv:2207.02749*.

Liu, L., Feng, S., Feng, Y., Zhu, X., and Liu, H. X. (2022). Learning-based stochastic driving model for autonomous vehicle testing. *Transportation research record*, 2676(1):54–64.

Liu, Y., Zhang, J., Fang, L., Jiang, Q., and Zhou, B. (2021). Multimodal motion prediction with stacked transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7577–7586.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. (2018). Microscopic traffic simulation using SUMO. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 2575–2582. IEEE.

Lu, Q., Tettamanti, T., Hörcher, D., and Varga, I. (2020). The impact of autonomous vehicles on urban traffic network capacity: an experimental analysis by microscopic traffic simulation. *Transportation Letters*, 12(8):540–549.

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

Madrigal, A. C. (2017). Inside Waymo's secret world for training self-driving cars. https://www.theatlantic.com/technology/archive/2017/08/inside-waymos-secret-testing-and-simulation-facilities/537648/.

Mahmassani, H. and Sheffi, Y. (1981). Using gap sequences to estimate gap acceptance functions. *Transportation Research Part B: Methodological*, 15(3):143–148.

Meng, Y., Qin, Z., and Fan, C. (2021). Reactive and safe road user simulations using neural barrier certificates. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6299–6306. IEEE.

Michigan Office of Highway Safety Planning (2022). Michigan Traffic Crash Facts. https://www.michigantrafficcrashfacts.org/.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106.

Mo, Z., Shi, R., and Di, X. (2021). A physics-informed deep learning paradigm for car-following models. *Transportation research part C: emerging technologies*, 130:103240.

Mooney, C. Z. (1997). *Monte carlo simulation*. Number 116. Sage.

National Highway Traffic Safety Administration (2017). Automated driving systems 2.0: a vision for safety. Technical report, United States. Department of Transportation.

National Highway Traffic Safety Administration (2022). 2020 fars/crss coding and validation manual. *Report No. DOT HS*.

Ngiam, J., Caine, B., Vasudevan, V., Zhang, Z., Chiang, H.-T. L., Ling, J., Roelofs, R., Bewley, A., Liu, C., Venugopal, A., et al. (2021). Scene transformer: A unified multi-task model for behavior prediction and planning. *arXiv e-prints*, pages arXiv–2106.

Nistér, D., Lee, H.-L., Ng, J., and Wang, Y. (2019). The safety force field. *NVIDIA White Paper*.

Nunes, A. and Axhausen, K. W. (2021). Road safety, health inequity and the imminence of autonomous vehicles. *Nature Machine Intelligence*, 3(8):654–655.

Nvidia Corporation (2021). NVIDIA DRIVE Sim. https://developer.nvidia.com/drive/simulation.

O'Kelly, M., Sinha, A., Namkoong, H., Tedrake, R., and Duchi, J. C. (2018). Scalable end-to-end autonomous vehicle testing via rare-event simulation. *Advances in neural information processing systems*, 31.

OpenStreetMap contributors (2017). Openstreetmap. https://planet.osm.org.

Owen, A. B. (2013). Monte carlo theory, methods and examples.

Pek, C., Manzinger, S., Koschi, M., and Althoff, M. (2020). Using online verification to prevent autonomous vehicles from causing accidents. *Nature Machine Intelligence*, 2(9):518–528.

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. (2018). Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE.

PTV GROUP (2018). PTV VISSIM 10 User Manual.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR.

Rhinehart, N., McAllister, R., Kitani, K., and Levine, S. (2019). Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2821–2830.

Richards, D. (2010). Relationship between speed and risk of fatal injury: pedestrians and car occupants.

Riedmaier, S., Ponn, T., Ludwig, D., Schick, B., and Diermeyer, F. (2020). Survey on scenario-based safety assessment of automated vehicles. *IEEE access*, 8:87456–87477.

SAE International (2021). Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. https://www.sae.org/standards/content/j3016_202104/.

Sangster, J., Rakha, H., and Du, J. (2013). Application of naturalistic driving data to modeling of driver car-following behavior. *Transportation research record*, 2390(1):20–33.

Sayer, J., LeBlanc, D., Bogard, S., Funkhouser, D., Bao, S., Buonarosa, M. L., Blankespoor, A., et al. (2011). Integrated vehicle-based safety systems field operational test: Final program report. Technical report, United States. Joint Program Office for Intelligent Transportation Systems.

Scanlon, J. M., Kusano, K. D., Daniel, T., Alderson, C., Ogle, A., and Victor, T. (2021). Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain. *Accident Analysis & Prevention*, 163:106454.

Schreier, M., Willert, V., and Adamy, J. (2016). An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2751–2766.

Schulz, J., Hubmann, C., Löchner, J., and Burschka, D. (2018). Interaction-aware probabilistic behavior prediction in urban environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3999–4006. IEEE.

Ścibior, A., Lioutas, V., Reda, D., Bateni, P., and Wood, F. (2021). Imagining the road ahead: Multi-agent trajectory prediction via differentiable simulation. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 720–725. IEEE.

Shah, S., Dey, D., Lovett, C., and Kapoor, A. (2018). AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer.

Shalev-Shwartz, S., Shammah, S., and Shashua, A. (2017). On a formal model of safe and scalable self-driving cars. *arXiv preprint arXiv:1708.06374*.

Shi, S., Jiang, L., Dai, D., and Schiele, B. (2022). Motion transformer with global intention localization and local movement refinement. *Advances in Neural Information Processing Systems*, 35:6531–6543.

Shladover, S. E. and Nowakowski, C. (2019). Regulatory challenges for road vehicle automation: Lessons from the california experience. *Transportation research part A: policy and practice*, 122:125–133.

Stern, R. E., Cui, S., Delle Monache, M. L., Bhadani, R., Bunting, M., Churchill, M., Hamilton, N., Pohlmann, H., Wu, F., Piccoli, B., Seibold, B., Sprinkle, J., and Work, D. B. (2018). Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C: Emerging Technologies*, 89:205–221.

Suo, S., Regalado, S., Casas, S., and Urtasun, R. (2021). Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10400–10409.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Swanson, E. D., Foderaro, F., Yanagisawa, M., Najm, W. G., Azeredo, P., et al. (2019). Statistics of light-vehicle pre-crash scenarios based on 2011–2015 national crash data. Technical report, United States. Department of Transportation. National Highway Traffic Safety Administration.

Taiebat, M., Brown, A. L., Safford, H. R., Qu, S., and Xu, M. (2018). A review on energy, environmental, and sustainability implications of connected and automated vehicles. *Environmental science & technology*, 52(20):11449–11465.

Talebpour, A., Mahmassani, H. S., and Hamdar, S. H. (2015). Modeling lane-changing behavior in a connected environment: A game theory approach. *Transportation Research Procedia*, 7:420–440.

Tesla, Inc. (2021). Tesla ai day. https://www.youtube.com/watch?v=j0z4FweCy4M.

Treiber, M., Hennecke, A., and Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805.

Treiber, M. and Kesting, A. (2017). The intelligent driver model with stochasticity-new insights into traffic flow oscillations. *Transportation research procedia*, 23:174–187.

United States. Department of Transportation (2018). Fatality Analysis Reporting System. https://www.nhtsa.gov/file-downloads?p=nhtsa/downloads/FARS/2018/National/.

Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

VISSIM (2012). Vissim 5.40-01, User Manual. Planung Transport Verkehr AG, Karlsruhe, Germany.

Wang, F., Li, L., Hu, J.-M., Ji, Y., Ma, R., and Jiang, R. (2009). A markov-process inspired ca model of highway traffic. *International Journal of Modern Physics C*, 20(01):117–131.

Wang, J., Wu, J., and Li, Y. (2015). The driving safety field based on driver–vehicle–road interactions. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2203–2214.

Wang, X., Jiang, R., Li, L., Lin, Y., Zheng, X., and Wang, F.-Y. (2017). Capturing car-following behaviors by deep learning. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):910–920.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR.

Waymo LLC (2021a). Simulation City: Introducing Waymo's most advanced simulation system yet for autonomous driving. https://blog.waymo.com/2021/06/SimulationCity.html.

Waymo LLC (2021b). Waymo safety report. https://downloads.ctfassets.net/e6t5diu0txbw/4mhzJxuCinbVNuyAKPPcOj/d1623d42ed7aaea46993c22ea7e50612/Waymo_Safety_Report_02-2021.pdf.

Webb, N., Smith, D., Ludwick, C., Victor, T., Hommes, Q., Favaro, F., Ivanov, G., and Daniel, T. (2020). Waymo's safety methodologies and safety readiness determinations. *arXiv preprint arXiv:2011.00054*.

Wikipedia contributors (2022). Hellinger distance.

Wu, C., Bayen, A. M., and Mehta, A. (2018). Stabilizing traffic with autonomous vehicles. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 6012–6018. IEEE.

Xie, D.-F., Fang, Z.-Z., Jia, B., and He, Z. (2019). A data-driven lane-changing model based on deep learning. *Transportation research part C: emerging technologies*, 106:41–60.

Xu, D., Chen, Y., Ivanovic, B., and Pavone, M. (2022). Bits: Bi-level imitation for traffic simulation. *arXiv preprint arXiv:2208.12403*.

Yang, H.-H. and Peng, H. (2010). Development of an errorable car-following driver model. *Vehicle System Dynamics*, 48(6):751–773.

Yang, Z., Chen, Y., Wang, J., Manivasagam, S., Ma, W.-C., Yang, A. J., and Urtasun, R. (2023). Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399.

Yang, Z., Feng, Y., and Liu, H. X. (2021). A cooperative driving framework for urban arterials in mixed traffic conditions. *Transportation research part C: emerging technologies*, 124:102918.

Yeo, H. (2008). *Asymmetric microscopic driving behavior theory*. University of California, Berkeley.

Yu, H., Jiang, R., He, Z., Zheng, Z., Li, L., Liu, R., and Chen, X. (2021). Automated vehicle-involved traffic flow studies: A survey of assumptions, models, speculations, and perspectives. *Transportation research part C: emerging technologies*, 127:103101.

Yunex Traffic Group (2022). Aimsun Next. https://www.aimsun.com/aimsun-next.

Zhang, K., Chang, C., Zhong, W., Li, S., Li, Z., and Li, L. (2022a). A systematic solution of human driving behavior modeling and simulation for automated vehicle studies. *IEEE Transactions on Intelligent Transportation Systems*.

Zhang, R., Zou, Z., Shen, S., and Liu, H. X. (2022b). Design, implementation, and evaluation of a roadside cooperative perception system. *Transportation Research Record*, page 03611981221092402.

Zhao, D., Lam, H., Peng, H., Bao, S., LeBlanc, D. J., Nobukawa, K., and Pan, C. S. (2016). Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques. *IEEE transactions on intelligent transportation systems*, 18(3):595–607.

Zhou, Z., Ye, L., Wang, J., Wu, K., and Lu, K. (2022). Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8823–8833.

Zhu, M., Wang, X., Tarko, A., et al. (2018a). Modeling car-following behavior on urban expressways in shanghai: A naturalistic driving study. *Transportation research part C: emerging technologies*, 93:425–445.

Zhu, M., Wang, X., and Wang, Y. (2018b). Human-like autonomous car-following model with deep reinforcement learning. *Transportation research part C: emerging technologies*, 97:348–368.

Zou, Z., Zhang, R., Shen, S., Pandey, G., Chakravarty, P., Parchami, A., and Liu, H. X. (2022). Real-time full-stack traffic scene perception for autonomous driving with roadside cameras. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 890–896. IEEE.