CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

UMTRI-2024-2
February 2024

Modeling Naturalistic Driving Environment with High-Resolution Trajectory Data

Shuo Feng, PhD
Xintao Yan, PhD
Henry Liu, PhD

UMTRI

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Report No. UMTRI-2024-2

October 2023

Project Start Date: April, 2022

Project End Date: March, 2023

Modeling Naturalistic Driving Environment with High-Resolution Trajectory Data

by

Shuo Feng, Assistant Research Scientist

Xintao Yan, Ph.D. Candidate

Henry Liu, Professor

University of Michigan

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

DISCLAIMER

Funding for this research was provided by the Center for Connected and Automated Transportation under Grant No. xxx of the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (OST-R), University Transportation Centers Program. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Suggested APA Format Citation:
Feng, S., Yan, X., & Liu, H.X. (2023). Modeling Naturalistic Driving Environment with High-Resolution Trajectory Data. Final Report. UMTRI-2024-2.
DOI: 10.7302/22417

Contacts

For more information:

Dr. Shuo Feng
University of Michigan
2350 Hayward, Ann Arbor, MI, 48109
Email: fshuo@umich.edu

Xintao Yan
University of Michigan
2350 Hayward, Ann Arbor, MI, 48109
Email: xintaoy@umich.edu

Dr. Henry X. Liu
University of Michigan
2901 Baxter Rd, Ann Arbor, MI, 48109
Phone: (734) 764-4354
Email: henryliu@umich.edu

Center for Connected and Automated Transportation
University of Michigan Transportation Research Institute
2901 Baxter Road
Ann Arbor, MI 48152
umtri-ccat@umich.edu
ccat.umtri.umich.edu
(734) 763-2498

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Technical Report Documentation Page

1. Report No. UMTRI-2024-2	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle Modeling Naturalistic Driving Environment with High-Resolution Trajectory Data DOI: 10.7302/22417		**5. Report Date** October 2023 **6. Performing Organization Code**
7. Author(s) Feng, Shuo, Ph.D., https://orcid.org/ 0000-0002-2117-4427 Yan, Xintao, https://orcid.org/0000-0002-0569-5628 Liu, Henry, Ph.D., https://orcid.org/0000-0002-3685-9920		**8. Performing Organization Report No.**
9. Performing Organization Name and Address UMTRI 2901 Baxter Road Ann Arbor, MI 48109		**10. Work Unit No.** **11. Contract or Grant No.** Contract No. 69A3551747105
12. Sponsoring Agency Name and Address Center for Connected and Automated Transportation University of Michigan Transportation Research Institute 2901 Baxter Road Ann Arbor, MI 48109		**13. Type of Report and Period Covered** Final Report April 2022 – March 2023 **14. Sponsoring Agency Code**

15. Supplementary Notes
Conducted under the U.S. DOT Office of the Assistant Secretary for Research and Technology's (OST-R) University Transportation Centers (UTC) program.

16. Abstract
For simulation to be an effective tool for the development and testing of autonomous vehicles, the simulator must be able to produce realistic safety-critical scenarios with distribution-level accuracy. However, due to the high dimensionality of real-world driving environments and the rarity of long-tail safety-critical events, how to achieve statistical realism in simulation is a long-standing problem. In this project, we develop NeuralNDE, a deep learning-based framework to learn multi-agent interaction behavior from high-resolution vehicle trajectory data, and propose a conflict critic model and a safety mapping network to refine the generation process of safety-critical events, following real-world occurring frequencies and patterns. The results show that NeuralNDE can achieve both accurate safety-critical driving statistics (e.g., crash rate/type/severity and near-miss statistics, etc.) and normal driving statistics (e.g., vehicle speed/distance/yielding behavior distributions, etc.), as demonstrated in the simulation of urban driving environments.

17. Key Words Autonomous vehicles, Simulation, Trajectory data	**18. Distribution Statement** No restrictions.		
19. Security Classif. (of this report) Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages** 42	**22. Price**

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

UMTRI · The University of Akron · Illinois Center for Transportation · Purdue University · Washtenaw Community College · Central State University

Table of Contents

List of Tables

No table of figures entries found.

List of Figures

Project Summary

This project develops a methodological framework for modeling the high-fidelity naturalistic driving environment (NDE) with high-resolution trajectory data. Different from traditional NDE models that mainly match the moments of macroscopic traffic behaviors, the high-fidelity NDE models will match the distributions of microscopic driving behaviors, which are critical for safety-critical applications such as autonomous vehicle testing and training. Specifically, we develop NeuralNDE, a deep learning-based model to learn multi-agent interaction behavior from vehicle trajectory data, and propose a conflict critic model and a safety mapping network to refine the generation process of safety-critical events, following real-world occurring frequencies and patterns. Utilizing high-resolution trajectory data collected by roadside sensors, we can train the NeuralNDE model to achieve statistical realism. The results show that NeuralNDE can achieve both accurate safety-critical driving statistics (e.g., crash rate/type/severity and near-miss statistics, etc.) and normal driving statistics (e.g., vehicle speed/distance/yielding behavior distributions, etc.), as demonstrated in the simulation of urban driving environments.

1. Introduction

Autonomous driving technologies are revolutionizing the future of transportation systems in unprecedented ways and speeds. However, safety remains the key challenge for the development and deployment of highly automated driving systems [1,2]. Simulation provides a controllable, efficient, and low-cost venue for both developing and testing autonomous vehicles (AV) [3,4]. But for simulation to be an effective tool, statistical realism of the simulated driving environment is a must [2,4,5,6,7]. In particular, the simulated environment needs to reproduce safety-critical encounters that AV might face in the real world with distribution-level accuracy. Unfortunately, the real-world naturalistic driving environment (NDE) is spatiotemporally complex and highly interactive. Therefore, how to achieve statistical realism for such simulators is a long-standing problem in the field.

In recent years, great efforts have been made in developing simulators for autonomous driving systems. 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 [8], Google/Waymo's CarCraft [9] and SimulationCity [5], Tesla's simulator, Microsoft's AirSim [10], NVIDIA's DRIVE Sim [11], Baidu's AADS [3], and Cruise's simulator, 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.

The key to high-fidelity NDE is accurate modeling of human driving behavior. Microscopic traffic simulators, which mimic the interactive agent behaviors through a combination of physics-driven models and hand-crafted rules, such as car-following models [12,13], lane-changing models [14,15], gap-acceptance models [16], etc., have been studied and developed in the transportation engineering domain for decades. Some well-known traffic simulators are SUMO [17], VISSIM [18], and AIMSUN [19]. Due to the limited capability of the underlying parametric models and manually encoded rules, the model fidelity is constrained. Many attempts have been made by using neural networks [20-25], Markovian-based models [6,26], Bayesian networks [27,28], and game theory [29], etc., to achieve better performance in modeling specific behaviors (e.g., car-following) or specific scenarios (e.g., unprotected left turn). However, they can hardly be generalized and scaled to model complex urban environments and highly interactive scenarios.

The focus of this study is to build a high-fidelity simulator that is statistically representative of

real-world driving environments, particularly for those long-tail safety-critical events. Especially, we aim to produce safety-critical events with distribution-level accuracy, including both crashes and near-misses, which are critical for training and testing AVs. This differentiates our proposed NeuralNDE model from most existing simulators based on imitation learning (including generative adversarial imitation learning) [30-36], where statistical realism is hardly considered and cannot be achieved. For example, the crash rates of these simulation environments are significantly higher (e.g., SimNet [33]) than that of real-world traffic. Moreover, these methods can only generate short-time simulations in the order of a few seconds (e.g., D2Sim [36]), which limits the capability of full-length trip training and evaluation of AVs. To reproduce high-fidelity safety-critical events, there are also methods proposed based on real-world event reconstruction. For example, the researchers constructed the simulation environment based on real-world fatal collision events from various data sources including police reports [37]. However, it may be difficult to reconstruct near-miss events using this method since the information needed for reconstruction is usually not available. Therefore, these reconstruction-based methods and our learning-based method serve to complement each other when building high-fidelity simulators.

The lack of statistical realism for simulation can potentially mislead AV development in both training and testing. An illustration example is shown in Figure 1a. 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 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 NDE distribution is extremely complex and in a very high-dimensional space as shown in Figure 1b. 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.

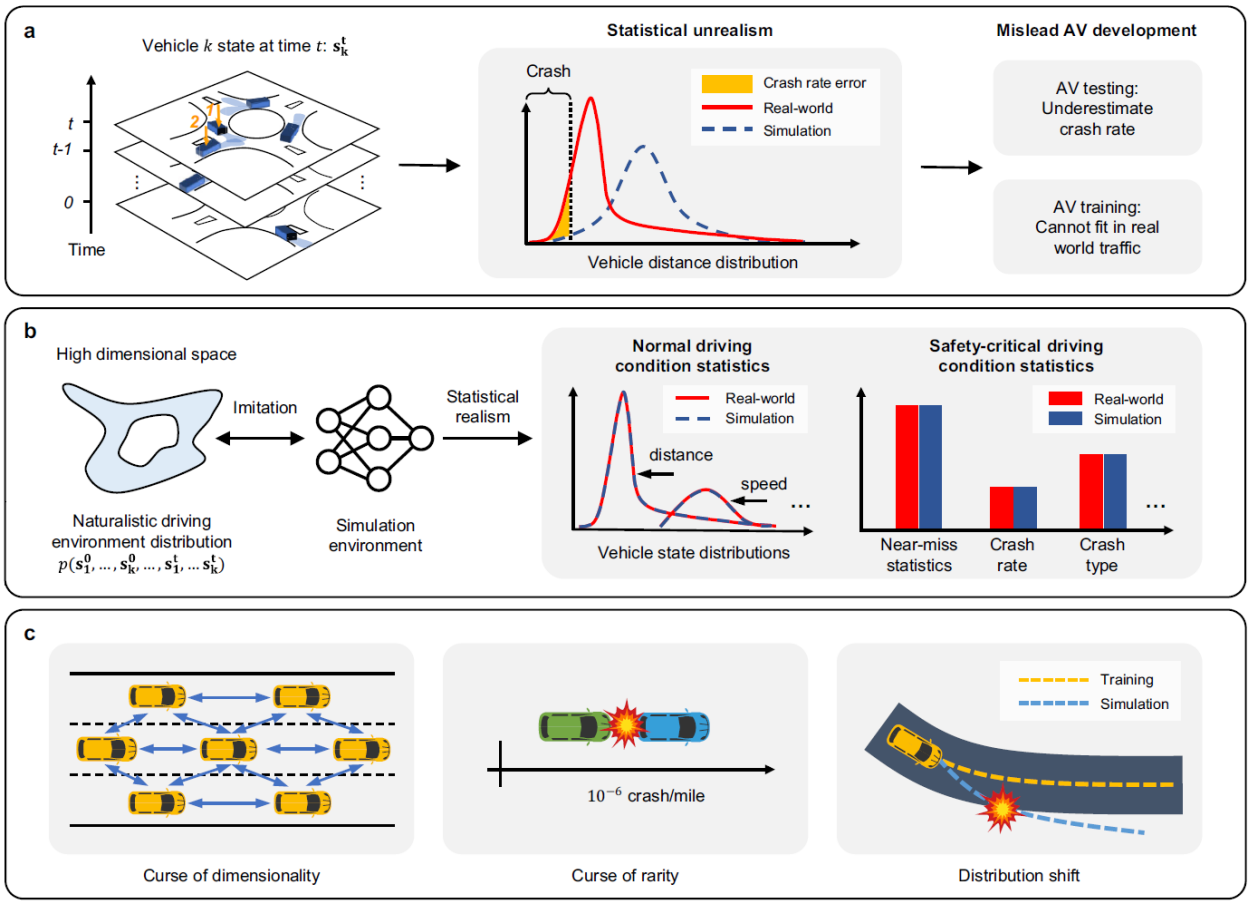

Figure 1 Modeling naturalistic driving environment with statistical realism. a, Statistical errors in simulation may mislead AV development. **b,** The underlying naturalistic driving environment distribution is highly complex and in a high dimensional space since it involves multiple agents and long time horizons. The simulation environment needs to achieve statistical realism, i.e., distribution-level accurate statistics regarding human driving behaviors in both normal and safety-critical driving conditions. **c,** Major challenges for modeling multi-agent interaction behaviors and constructing naturalistic driving environments. The challenges include the "curse of dimensionality" for multi-agent highly interactive behaviors, the "curse of rarity" [8] of safety-critical events in the real world, and the "distribution shift" for long-time simulations.

The challenges of modeling NDE with statistical realism mainly come from three aspects as shown in Figure 1c. 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 make NDE modeling a very high-dimensional problem. The second challenge is from the "curse of rarity" [38]. 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 [39]), 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 [38]. The third challenge is from the "distribution shift" [30], which is particularly critical for learning-based simulators. 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.

In this project, we solve this long-standing problem by developing NeuralNDE - a novel deep learning-based framework for simulating Naturalistic Driving Environment with statistical realism. To demonstrate the effectiveness of our approach, we construct a multi-lane roundabout environment located in the US using real-world data. The simulated environment is validated to be statistically accurate with the real world, including vehicle instantaneous speed, distance, and yielding behavior. More importantly, the proposed NeuralNDE can achieve accurate safety-critical statistics including both crash and near-miss measurements, for example, crash rate, crash type, crash severity, post-encroachment time (PET [44]), etc. The fidelity of NeuralNDE-generated crash events is further validated against real-world crash videos and police crash reports. To the best of our knowledge, this is the first time that a simulation environment can systematically reproduce the real-world driving environment with statistical realism, particularly for those long-tail safety-critical events that are critical to AV safety. In addition, the proposed environment can perform long-time (hour-level) simulation, where the AV under training or testing can continuously interact with background vehicles. The proposed NeuralNDE should be readily integrated with different high-fidelity AV simulators, for example, CARLA [5], which focuses on photorealistic rendering and sensor simulations, to provide a realistic traffic environment. Furthermore, it should be noted that the proposed NeuralNDE model can be used for other safety-related applications other than AV training and testing. For example, the proposed NeuralNDE model can be used to estimate the safety performance of a traffic facility under different traffic flow conditions.

2. Methodology

2.1 Overall framework

The overview of the proposed framework is shown in Figure 2. We frame the simulation modeling under an imitation learning 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., GTP [40] and BERT [41])

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 to simulate the vehicle state at future moments. To further overcome the distribution shift issue, we integrate the generative adversarial training as in in GAN [42] and GAIL [43], 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.

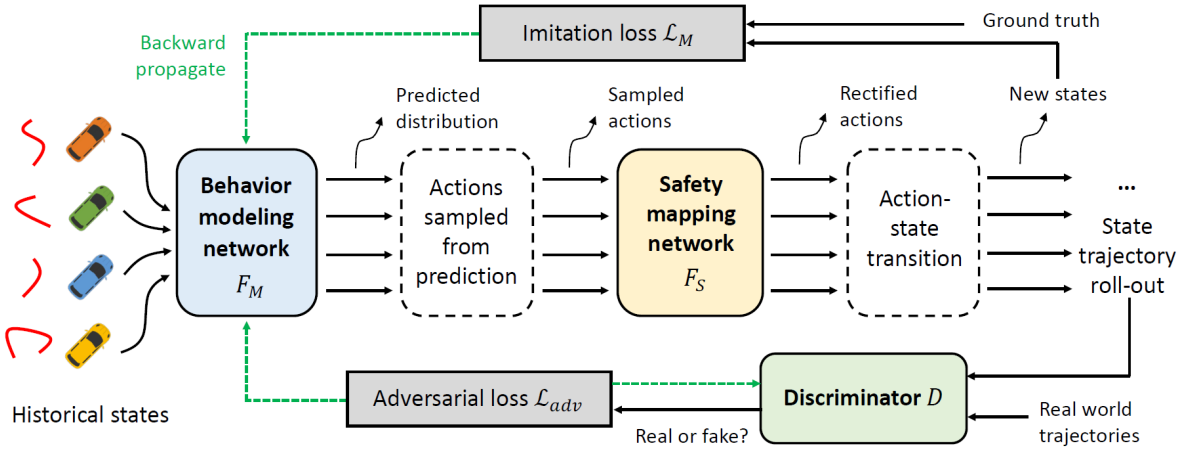

Figure 2 The framework and training pipeline of the NeuralNDE.

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 during the inference time as shown in Figure 3. 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.

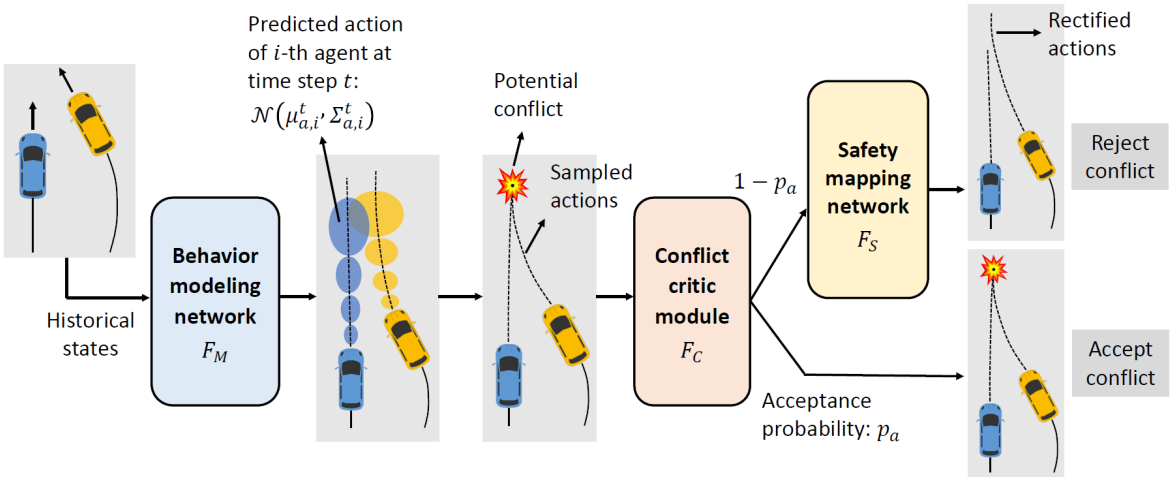

Figure 3 Demonstration of the behavior modeling network, conflict critic module, and safety mapping network during the inference time.

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 4, 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.

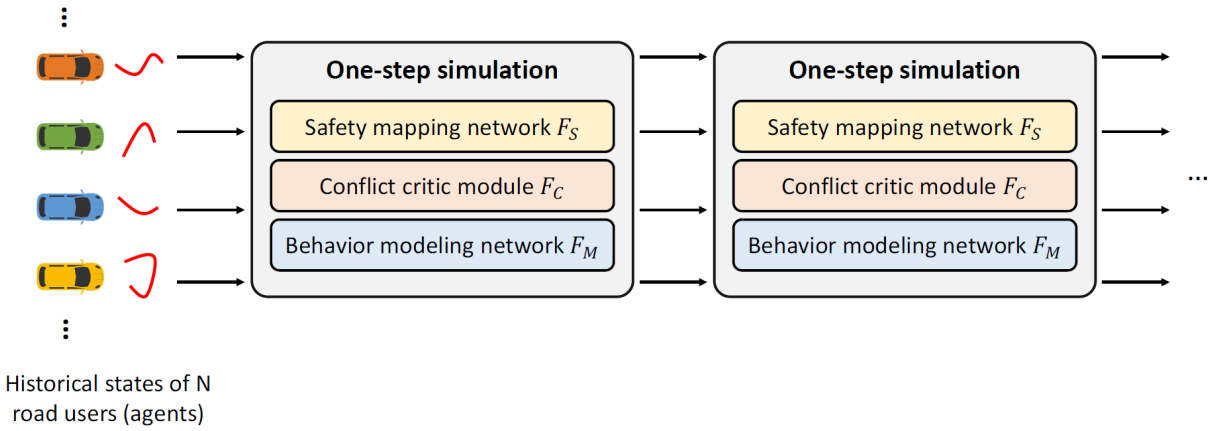

Figure 4 Illustration of the simulation process.

2.2 Behavior modeling network

We frame the behavior modeling via imitation learning 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 $\mathbf{s_i^t}$ and $\mathbf{a_i^t}$ represent the state vector (e.g., location, pose, vehicle size, etc.) and action vector (e.g., acceleration, yaw rate, etc.) of i th agent at time step t . $\mathbf{S_N^{\tau:t}} = \{\mathbf{s_1^{t-\tau+1}}, \dots \mathbf{s_1^t}, \dots \mathbf{s_N^{t-\tau+1}}, \dots \mathbf{s_N^t}\}$ 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 $p(\mathbf{a_1^t}, \dots, \mathbf{a_N^t} | \mathbf{S_N^{\tau:t}})$ 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 [54], and have revealed remarkable performance in many applications, including computer vision [55], bioinformatics [56], and multimodal data generative modeling [57].

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. In this study, our model can handle up to 32 objects simultaneously, considering the size of the roundabout and for the convenience of experiments. However, the framework we designed should be able to handle a much larger number of objects, as each agent in the simulation is modeled as a "token". It has been shown in previous studies [58,59] that the Transformer can handle up to thousands of "tokens". 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 $\mathrm{F_M}$ takes in the historical states $\mathbf{S_N^{\tau:t}}$ of all agents and is trained to jointly predict their future actions $(\mathbf{a_1^t}, \dots, \mathbf{a_N^t})$. Instead of predicting deterministic actions values, we predict the stepwise action distributions and consider distribution as a multi-variable Gaussian over their action space:

$$p(\mathbf{S_N^{\tau:t}}) = \mathrm{F_M}(\mathbf{S_N^{\tau:t}}) \sim N(\boldsymbol{\mu}^{\mathbf{t}}_{\mathbf{a,i=1\dots N}}, \boldsymbol{\Sigma}^{\mathbf{t}}_{\mathbf{a,i=1\dots N}}), \quad (1)$$

where $p(\mathbf{S_N^{\tau:t}})$ is the joint action distribution and $\boldsymbol{\mu}^{\mathbf{t}}_{\mathbf{a,i=1\dots N}}$ and $\boldsymbol{\Sigma}^{\mathbf{t}}_{\mathbf{a,i=1\dots N}}$ 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:

$$\mathbf{a_1^t}, \dots, \mathbf{a_N^t} \leftarrow N(\boldsymbol{\mu}^{\mathbf{t}}_{\mathbf{a,i=1\dots N}}, \boldsymbol{\Sigma}^{\mathbf{t}}_{\mathbf{a,i=1\dots N}}). \quad (2)$$

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

$$\mathbf{s_i^{t+1}} = \mathrm{T}(\mathbf{a_i^t}, \mathbf{s_i^t}). \quad (3)$$

The above processing will be repeated so that new states of all agents can be generated in an auto-regressive manner. In practice, instead of one-step prediction, multiple time steps (e.g., κ steps) predictions $\mathbf{S_N^{t:\kappa}} = \{\mathbf{s_1^{t+1}}, \dots \mathbf{s_1^{t+\kappa}}, \dots \mathbf{s_N^{t+1}}, \dots \mathbf{s_N^{t+\kappa}}\}$ 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 agents' 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 [60].

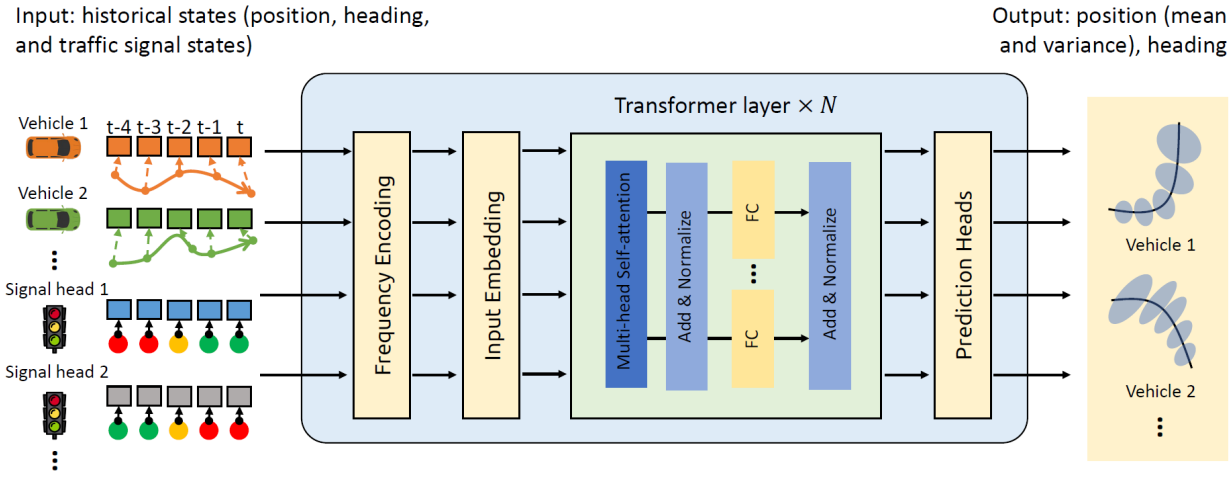

Figure 5 Network architecture of the behavior modeling network.

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 5. Detailed network architecture can be found in the later section. The input embedding layer is a fully

connected layer with weights shared across different tokens. The Transformer backbone consists of several standard BERT [41] 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 Mildenhall et al. [61] 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 $\mathbf{s}$ after mapping can be written as follows:

$$\boldsymbol{\gamma}(\boldsymbol{s}) = [\mathbf{s}, \sin(2^0\pi\mathbf{s}), \cos(2^0\pi\mathbf{s}), \dots, \sin(2^{L-1}\pi\mathbf{s}), \cos(2^{L-1}\pi\mathbf{s})]. \quad (4)$$

To train the model $\mathrm{F_M}$ with implicit variance, in the prediction layer, two prediction heads are attached for each input token at the output end, one for predicting $\boldsymbol{\mu}_{\mathbf{a,i}}^{\mathbf{t}}$, another for predicting $\boldsymbol{\Sigma}_{\mathbf{a,i}}^{\mathbf{t}}$. 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 ($t = 1, \dots, T$), then the likelihood function can be written as follows:

$$p(\mathrm{F_M}) = p(\mathbf{s_1^1}, \mathbf{s_1^2}, \dots, \mathbf{s_i^t}, \dots, \mathbf{s_N^T}). \quad (5)$$

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., $\boldsymbol{\Sigma}_{\mathbf{a,i}}^{\mathbf{t}} \approx diag(\sigma_{i,1}^t, \dots, \sigma_{i,D}^t)$, D is the dimension of the action vector. Then the joint probability of an action vector $\mathbf{a_i^t}$ can be implied as follows:

$$p(\mathbf{s}^{\mathbf{t}}) = \prod_{j=1}^{D} \frac{1}{(2\pi)^{\frac{1}{2}}} \frac{1}{\sigma_{i,j}^t} exp\{-\frac{(\mu_{i,j}^t - a_{i,j}^t)^2}{2(\sigma_{i,j}^t)^2}\}, \quad (6)$$

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

We approximate the joint probability distribution $p(\mathrm{F_M})$ in Equation (5) as the multiplicative form of each agent's marginal probabilities, and combine it with Equation (6) to derive the loss function in the negative log-likelihood form as follows:

$$\mathrm{L_M(F_M)} = \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{D}[\ln(\sigma_{i,j}^t) + \frac{(\mu_{i,j}^t - a_{i,j}^t)^2}{2(\sigma_{i,j}^t)^2}]. \quad (7)$$

The approximation of Equation (5) will not affect the solution since the term $\frac{1}{2}(\mu_{i,j}^t - a_{i,j}^t)^2/(\sigma_{i,j}^t)^2$ in Equation (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.

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 $\mathrm{F_c}$ to control the occurring frequencies and patterns of safety-critical behaviors during the inference time to achieve statistical realism. The input to $\mathrm{F_c}$ are the sampled κ steps predicted trajectories of all N agents $\mathbf{S_N^{t:\kappa}} = \{\mathbf{s_1^{t+1}}, \dots \mathbf{s_1^{t+\kappa}}, \dots \mathbf{s_N^{t+1}}, \dots \mathbf{s_N^{t+\kappa}}\}$ generated by the behavior modeling network at the current time t. The output of $\mathrm{F_c}$ is the acceptance probability p_a for not passing through the safety mapping network:

$$p_a = \mathrm{F_c}(\mathbf{S_N^{t:\kappa}}). \tag{8}$$

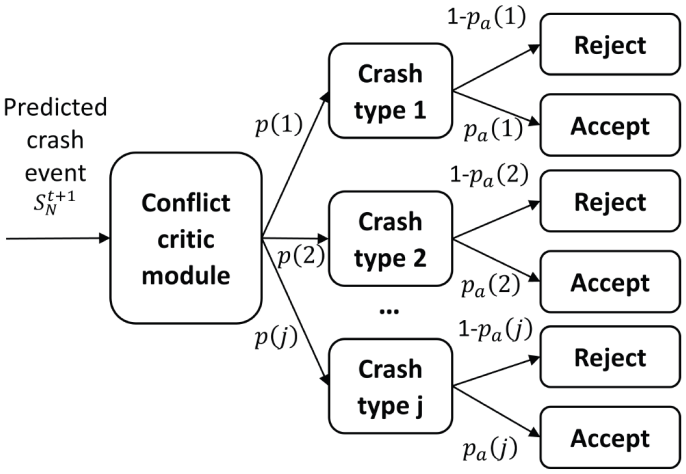

Figure 6 Illustration of the conflict critic module.

If there is a potential conflict in predicted trajectories $\mathbf{S_N^{t:\kappa}}$, 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 $\mathrm{F_c}$ 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 study, each crash type will have a specific acceptance probability. The implementation details and calibration methods are introduced in the following

paragraphs.

In this project, we consider vehicle conflicts in a one-step prediction for simplicity. Let $\mathbf{S_N^{t+1}}$ denotes 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 $\mathbf{S_N^{t+1}}$ 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 $\mathbf{S_N^{t+1}}$, the acceptance probability will be zero. An illustration figure is shown in Figure 6. 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 first 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 first iteration c^1. Then linearly update the uniform acceptance probability as follows

$$p_{ua}^{i+1} = c^{gt} \cdot \frac{p_{ua}^i}{c^i}, \tag{9}$$

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 10) and crash type distribution (Equation 11):

$$\sum_j p(j) p_a(j) = p_{ua}. \tag{10}$$

$$\frac{p(j)p_a(j)}{\sum_j p(j)p_a(j)} = c^{gt}(j), \forall j \in J, \tag{11}$$

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 (12), the summation of $p(j)p_a(j)$ over all potential crash types j is 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

$$p_a(j) = p_{ua} \cdot \frac{c^{gt}(j)}{p(j)}. \tag{12}$$

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

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 $\{\mathbf{S_N^t}, \mathbf{A_N^{t:\kappa}}\}$, the safety mapping network $\mathrm{F_S}$ jointly predicts the rectified actions $\mathbf{A_N^{t:\kappa,*}}$ of all agents as follows

$$\mathbf{A_N^{t:\kappa,*}} = \mathrm{F_S}(\mathbf{S_N^t}, \mathbf{A_N^{t:\kappa}}). \tag{13}$$

If there is an impending crash using the original action vector $\mathbf{A_N^{t:\kappa}}$, 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.

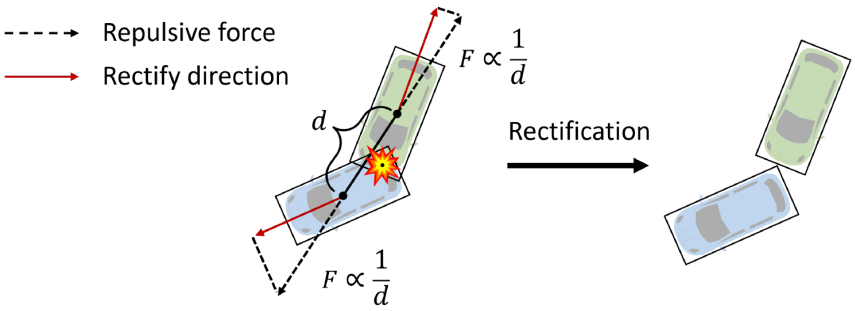

Figure 7 Illustration figure of the physics-based safety mapping rule to guide vehicle behavior in safety-critical situations.

The safety mapping network is trained to imitate existing model-based safety guards based on domain knowledge. In this project, 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 7. 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:

$$L_S(F_S) = ||F_S(S_N^t, A_N^t) - \hat{A}_N^t||_1, \quad (14)$$

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 2.

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 [62], potential force field-based methods [63,64], online verification methods [65], etc.

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 [42]. 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 multi-layer perceptron 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, $\hat{S}\sim p_R(S_N^t)$ represents a trajectory sampled from the real-world data distribution, and $S\sim p_G(S_N^t)$ represents a trajectory generated from the simulation. We follow a standard adversarial training pipeline and define the adversarial loss functions as follows:

$$L_{adv}(F_M, D) = E_{\hat{S}\sim p_R(S_N^t)}[\log D(\hat{S})] + E_{S\sim p_G(S_N^t)}[\log(1 - D(S))]. \quad (15)$$

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 (9)) of the behavior modeling network F_M, our final objective function is defined as follows:

$$F_M^*, D^* = \arg\min_{F_M}\max_{D}\{L_M(F_M) + \beta L_{adv}(F_M, D)\}. \quad (16)$$

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. Implementation details

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 8. This is a busy roundabout with a large traffic volume and the fourth highest crash rate in Michigan [45]. A roadside perception system [46,47] 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 used crash data from large-scale trajectories and police crash reports [50] to obtain ground-truth safety-critical events statistics

(e.g., crash rate and crash type distribution).

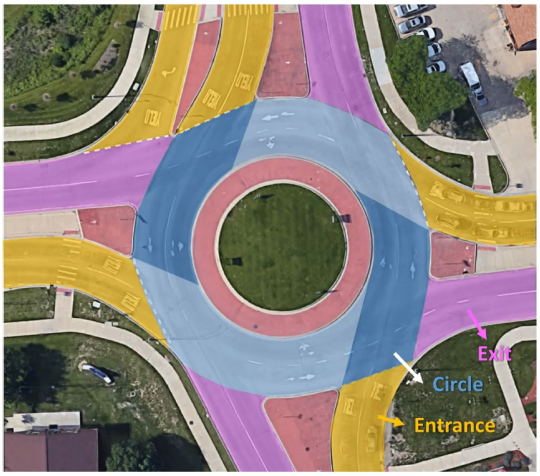

Figure 8 Illustration figure of the roundabout at Ann Arbor, Michigan, USA.

3.2 Network architecture

The network architecture of the behavior modeling network is shown in Figure 5. Each road user is considered as a token and the input to the network is its historical trajectory, i.e., position (x, y coordinates) and heading (cosine and sine of heading), of all road users within the historical time window. 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 [61] layer, in which we apply a set of sine and cosine basis functions that project 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 study. 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 [41] 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*512*256*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 Training details

We train the safety mapping network by using the RMSprop 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 7. 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 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.

4. Case studies

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 timestep, 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. 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 approximately 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 1,000 cores and 2000 GB RAM. It took around 1,440 seconds of real-world time to conduct 3,600 seconds of simulation. Therefore, the simulation speed ratio (simulation time/real-world time) is around 0.4.

We compare the proposed method with SUMO [17] - 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. 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 [13] and SL2015 [17] model are used as the car-following model and lane-changing model, respectively.

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. Vehicle crash rate;
5. Vehicle crash type distribution;
6. Vehicle crash severity distribution;
7. 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 9. 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 project.

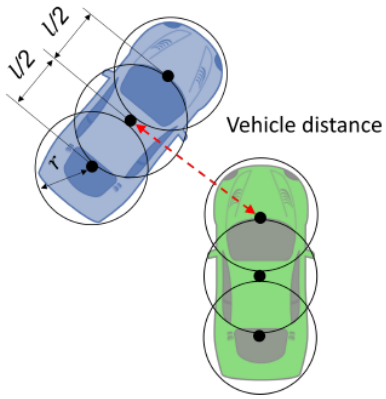

Figure 9 Illustration figure of the vehicle distance.

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 10. Vehicles in the corresponding circle quadrant as shown in Figure 10 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.

Figure 10 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 NHTSA [49].

We use the 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. Many existing studies investigated the relationship between Delta-V and occupant injury level, we follow their found thresholds 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 $(34, \infty)$.

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 KL-

divergence are used as measurements. For two discrete probability distributions $\mathbf{P}$ and $\mathbf{Q}$, their Hellinger distance D_H is calculated as follows:

$$D_H(\mathbf{P},\mathbf{Q}) = \frac{1}{\sqrt{2}}\sqrt{\sum_x (\sqrt{P(x)} - \sqrt{Q(x)})^2}, \quad (17)$$

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 $\mathbf{P}$ is the real-world data distribution and $\mathbf{Q}$ is the simulated distribution, the KL-divergence D_{KL} can be calculated as

$$D_{KL}(\mathbf{P},\mathbf{Q}) = \sum_x P(x) \log \frac{P(x)}{Q(x)}. \quad (18)$$

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

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 11a. 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 11b, reflecting the full distribution of encounters that AV might face in the real world.

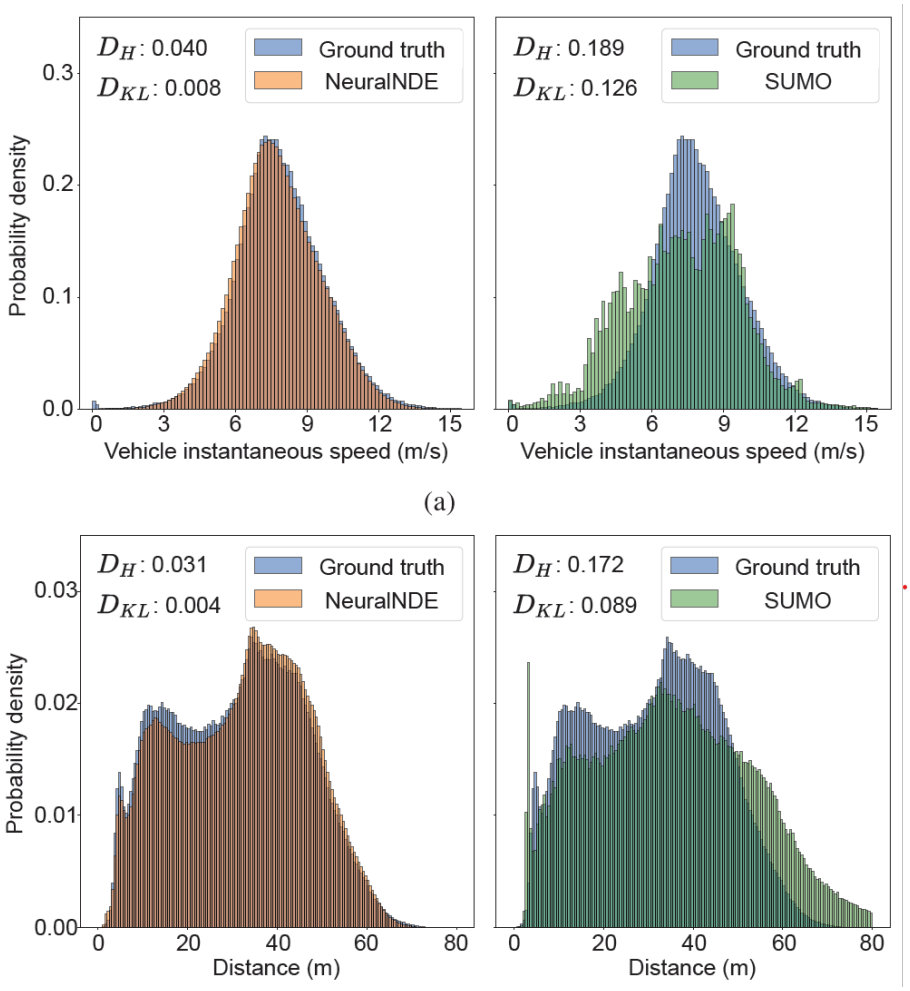

Figure 11 Statistical realism of normal driving behavior. a, Vehicle instantaneous speed distribution. **b,** Vehicle distance distribution.

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 12a and Figure 12b, respectively. We can find that NeuralNDE can accurately replicate human yielding behavior and significantly outperforms the SUMO simulator. Human drivers are naturally heterogeneous and have different characteristics. 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.

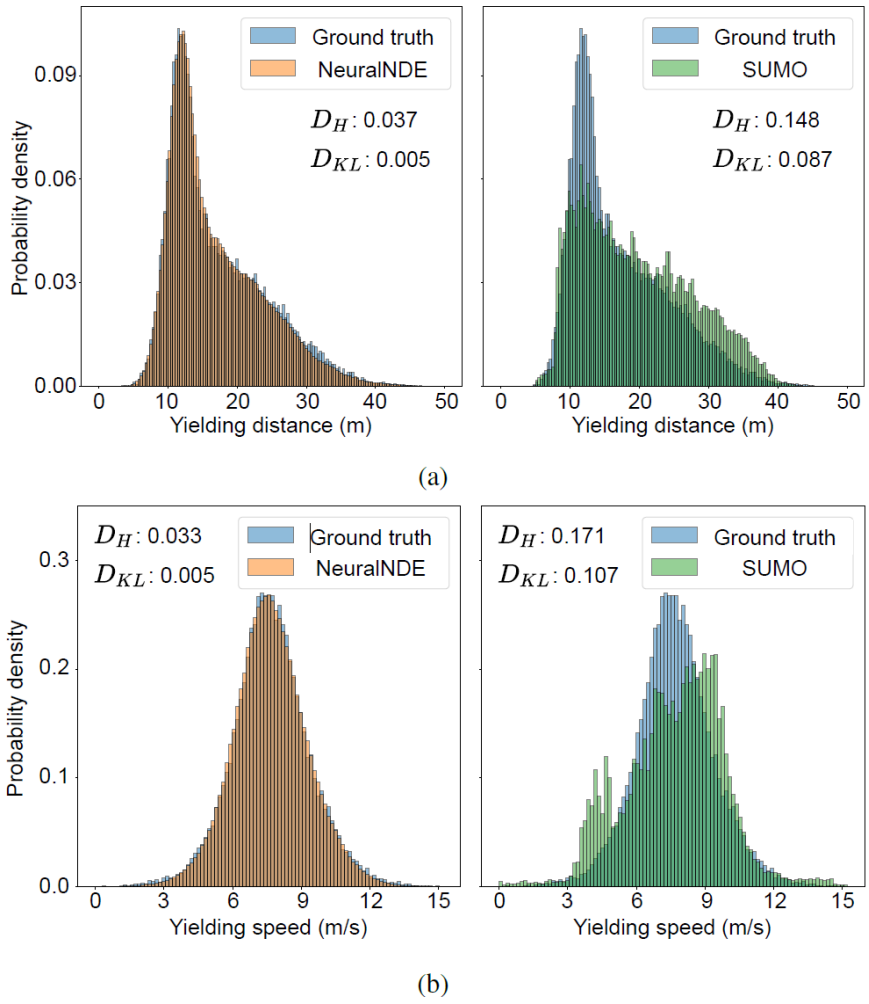

Figure 12 Statistical realism of normal driving behavior. a, Yielding distance distribution: distance between the yielding vehicle and its nearest conflicting vehicle. **b,** Yielding speed distribution: speed of the nearest conflicting vehicle.

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 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 ground-truth crash rate of the studied roundabout 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 13a and Figure 13b, respectively. The ground-truth crash type and crash severity distributions are queried from the Michigan Traffic Crash Facts dataset [50] 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 [30,33,34,35,51], 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.

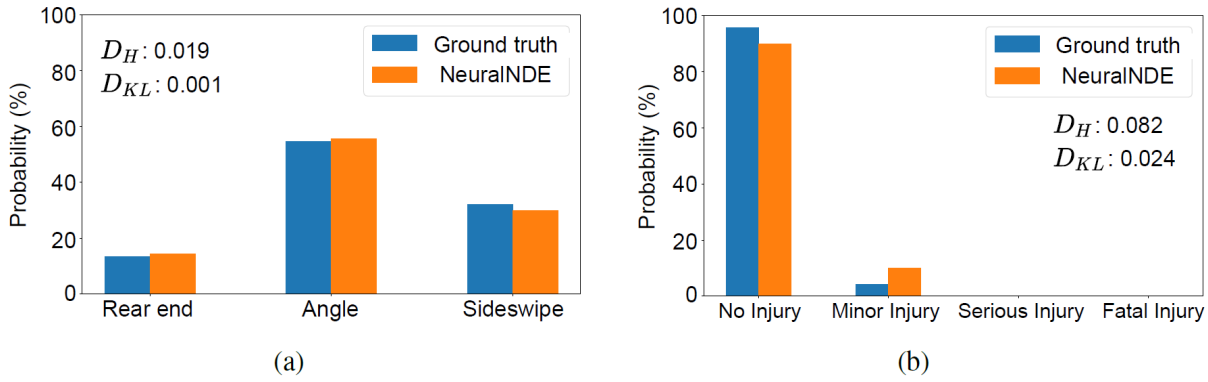

Figure 13 Statistical realism of safety-critical driving behavior. a, Vehicle crash type distribution. **b,** Vehicle crash severity distribution.

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 14a. We can find that NeuralNDE can replicate the distance in near-miss situations with high accuracy. Similarly, the simulated PET distribution can also accurately reproduce real-world dangerous driving conditions as shown in Figure 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.

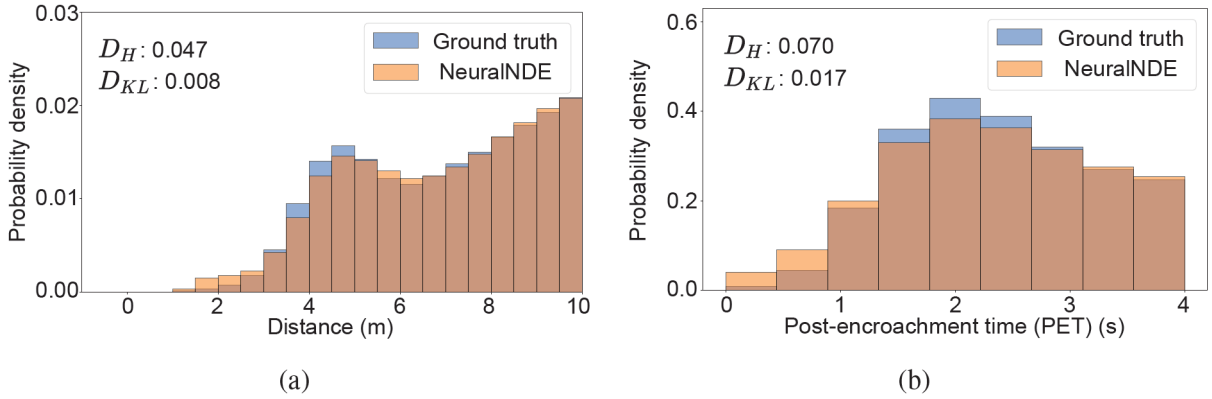

Figure 14 Statistical realism of safety-critical driving behavior. a, Vehicle distance distribution in near-miss situations. **b,** Post-encroachment time (PET) distribution in near-miss situations.

4.5 Generated crash events

The proposed NeuralNDE can generate complex and diverse interactions that happen in real-world traffic. Human drivers naturally exhibit different characteristics and spontaneously interact, negotiate, and cooperate to navigate through the roundabout. 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 16.

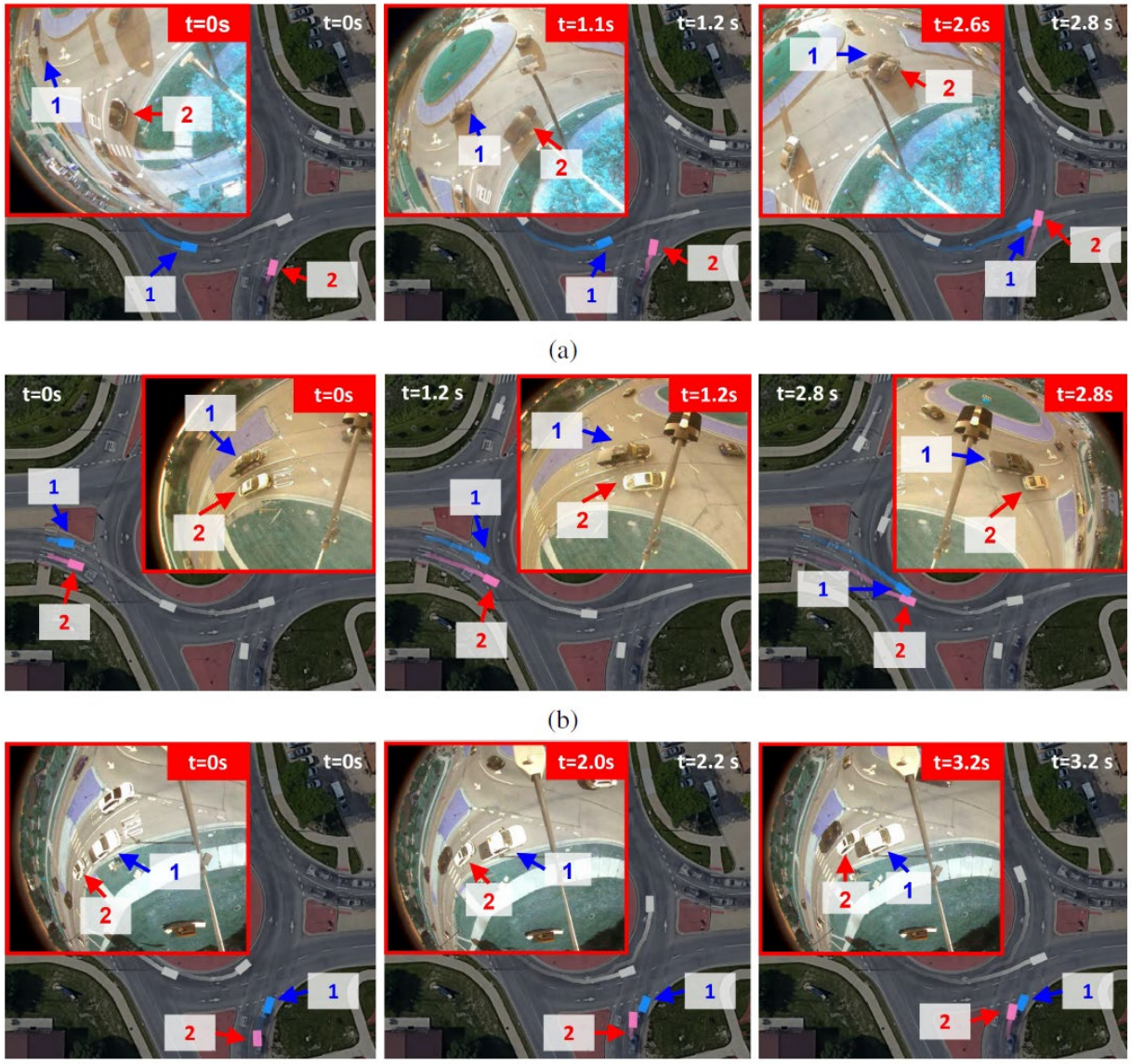

Figure 15 Crash events in the real world and NeuralNDE. a, Angle crash caused by failure to yield. **b,** Sideswipe crash caused by improper lane usage. **c,** Rear-end crash caused by failure to stop within assured clear distance.

The first case is an angle crash caused by failure to yield as shown in Figure 16a, 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 world as shown by the images in the red box of Figure 16a. 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 16b. 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 16b, 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 16c. 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.

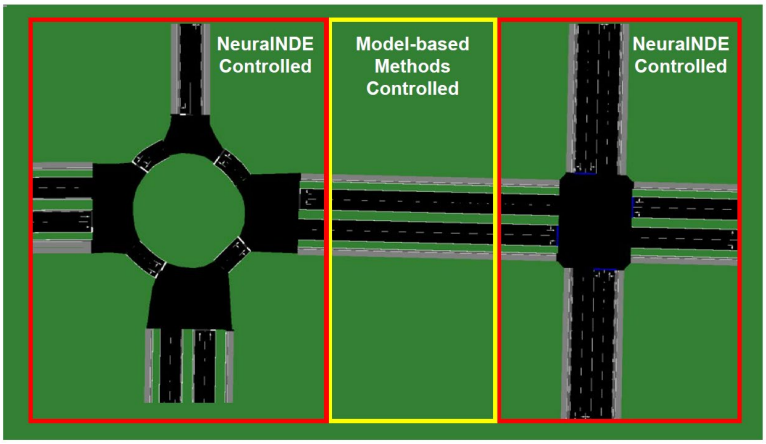

Figure 16 Proof-of-concept for modeling a road network.

4.6 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 16, that involves two scenarios, i.e., a four-way stop sign-controlled intersection and a two-lane roundabout. We use SUMO [17] 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 are controlled by trained NeuralNDE models, and the transition areas between the two scenarios are controlled by rule-based IDM car-following model [13] and SL2015 [17] lane-changing model.

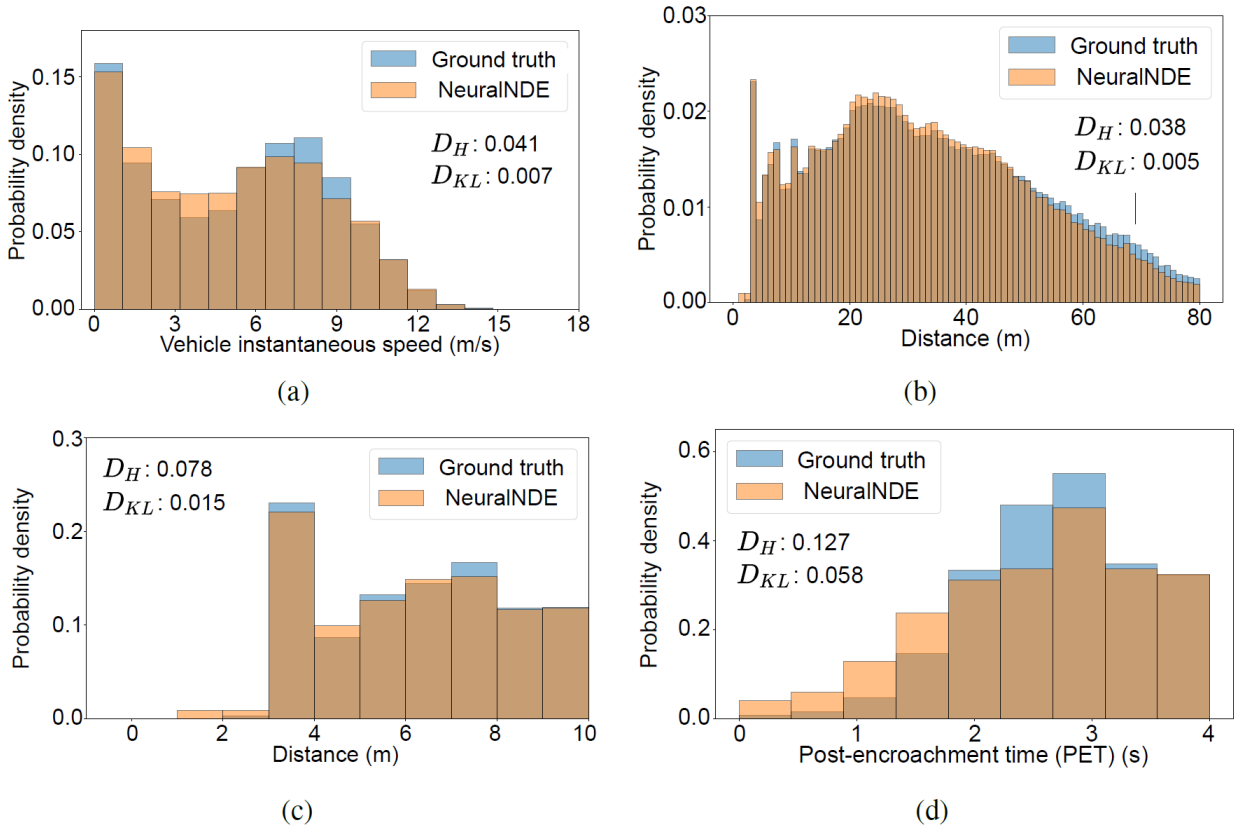

Figure 17 Statistical realism of the intersection area in the road network. a, Vehicle instantaneous speed distribution. **b,** Vehicle distance distribution. **c,** Vehicle distance

distribution in near-miss situations. **d,** Post-encroachment time (PET) distribution in near-miss situations.

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 run 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 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 17c-d, respectively. We can find that the proposed method can replicate the ground-truth with high accuracy.

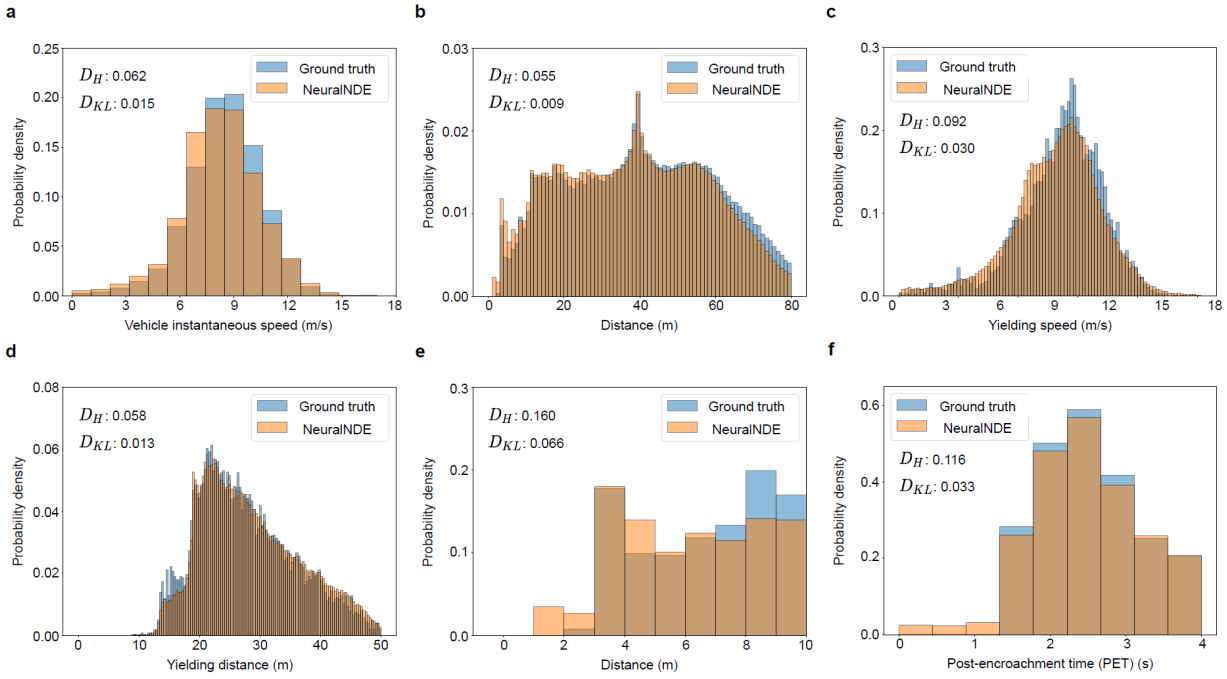

Figure 18 Statistical realism of the roundabout area in the road network. a, Vehicle instantaneous speed distribution. **b,** Vehicle distance distribution. **c,** Yielding speed distribution. **d,** Yielding distance distribution. **e,** Vehicle distance distribution in near-miss situations. **f,** Post-encroachment time (PET) distribution in near-miss situations.

For the roundabout area, the vehicle instantaneous speed, distance, yielding speed, and yielding distance results are shown in Figure 18a-d, respectively. The safety-critical events metrics (vehicle distance in the near-miss situations and PET) are shown in Figure 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.

5. Findings

In this project, we demonstrated the effectiveness of NeuralNDE for modeling a complex urban driving environment with statistical realism for both normal and safety-critical driving conditions. To the best of the authors' knowledge, this is the first time that a simulation environment can be statistically representative of the real-world driving environment. 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 [8]) 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.

6. Recommendations

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 [53], further development might be required to consider the AV influences on surrounding human-driven vehicles.

7. Outputs, Outcome, and Impacts

Outcome: In this project, we develop methodologies for building high-fidelity simulation environments, which serve as the prerequisite and foundation of various AV applications, for example, training and testing the AV safety performance. We leverage the recent advances in fundamental models and use Transformer as the backbone to model human interaction behaviors. The proposed method can reproduce the real-world driving environment with statistical realism, particularly for safety-critical situations. With such high-fidelity simulator, we can obtain trustworthy results with minimum sim-to-real gap.

Impact: The proposed method paves 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. Beyond its applications in the context of AVs, the proposed method 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.

Outputs: The following outputs were generated during the performance of this project:

- Journal Paper [66]: Yan, X., Zou, Z., Feng, S., Zhu, H., Sun, H., & Liu, H. X. (2023). Learning naturalistic driving environment with statistical realism. Nature Communications, 14(1), 2037. https://www.nature.com/articles/s41467-023-37677-5
- Conference oral presentation:
    - 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2023). Session: Autonomous Vehicles - Safety and Systems.
    - 2023 INFORMS Annual Meeting. Session: The Impact of Uncertainty on Transportation Systems.
- Conference poster: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2023). Session: Late Breaking Posters.
- Media coverage: Ann Arbor Observer October 2023 issue. Also available digitally: https://annarborobserver.com/virtual-roundabout/
- Learning Naturalistic Driving Environment Code. DOI: 10.7302/22417
- Learning Naturalistic Driving Environment Data. DOI: 10.7302/22417

References

1. Safe driving cars. *Nat. Mach. Intell*, **4**, 95-96 (2022).
2. Feng, S., Sun, H., Yan, X., Zhu, H., Zou, Z., Shen, S., & Liu, H. X. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*. **615**, 620-627 (2023).
3. Li, W. et al. AADS: Augmented autonomous driving simulation using data-driven algorithms. *Sci. Robot.* **4**, eaaw0863 (2019).
4. Feng, S., Yan, X., Sun, H., Feng, Y., & Liu, H. X. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nat. Commun.* **12**, 1-14 (2021).
5. Simulation City: Introducing Waymo's most advanced simulation system yet for autonomous driving. https://blog.waymo.com/2021/06/SimulationCity.html (2021).
6. Yan, X., Feng, S., Sun, H., & Liu, H. X. Distributionally consistent simulation of naturalistic driving environment for autonomous vehicle testing. Preprint at https://arxiv.org/abs/2101.02828 (2021).
7. Langer, M., Kates, R., & Bogenberger, K. Simulation of Urban Crash Occurrence Based on Real-World Crash Data. *Transport. Res. Rec.* (2022).
8. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. CARLA: An open urban driving simulator. *Conference on robot learning* (PMLR) 1-16 (2017).
9. Madrigal, A. C. Inside Waymo's Secret World for Training Self-Driving Cars. *Atlatic* **23**, 3–1 (2017).
10. Shah, S., Dey, D., Lovett, C. & Kapoor, A. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In (Hutter, M. & Siegwart, R. eds) *Field and Service Robotics* (Springer, Cham 2017).
11. NVIDIA. NVIDIA DRIVE Sim. https://developer.nvidia.com/drive/drive-sim (2021).
12. Bando, M., Hasebe, K., Nakayama, A., Shibata, A., & Sugiyama, Y. Dynamical model of traffic congestion and numerical simulation. *Phys. Rev. E.* **51**, 1035 (1995).
13. Treiber, M., Hennecke, A., & Helbing, D. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E*. **62**, 1805 (2000).
14. Kesting, A., Treiber, M., & Helbing, D. General lane-changing model MOBIL for car-following models. *Transport. Res. Rec.* **1999**, 86-94 (2007).
15. Erdmann, J. SUMO's lane-changing model. In *Modeling Mobility with Open Data*. (Springer, Cham 2015).
16. Mahmassani, H., & Sheffi, Y. Using gap sequences to estimate gap acceptance functions. *Transp. Res. B. Methodol*. **15**, 143-148 (1981).
17. Lopez, P. et al. Microscopic traffic simulation using SUMO. *International Conference on Intelligent Transportation Systems (ITSC)* 2575-2582 (IEEE, 2018).
18. VISSIM. Vissim 5.40-01, user manual. Planung Transport Verkehr AG, Karlsruhe, Germany (2012).
19. AIMSUN. AIMSUN Next. https://www.aimsun.com/aimsun-next (2022).

20. Wang, X. et al. Capturing car-following behaviors by deep learning. *IEEE Trans. Intell. Transp. Syst.* **19**, 910-920 (2017).
21. Zhu, M., Wang, X., & Wang, Y. Human-like autonomous car-following model with deep reinforcement learning. *Transp. Res. C. Emerg. Technol.* **97**, 348-368 (2018).
22. Xie, D., Fang, Z., Jia, B., & He, Z. A data-driven lane-changing model based on deep learning. *Transp. Res. C. Emerg. Technol.* **106**, 41-60 (2019).
23. Mo, Z., Shi, R., & Di, X. A physics-informed deep learning paradigm for car-following models. *Transp. Res. C. Emerg. Technol.* **130**, 103240 (2021).
24. Liu, L., Feng, S., Feng, Y., Zhu, X., & Liu, H. X. Learning-based stochastic driving model for autonomous vehicle testing. *Transport. Res. Rec.* **5676**, 54-64 (2022).
25. Liu, H., Tian, Y., Sun, J., & Wang, D. An exploration of data-driven microscopic simulation for traffic system and case study of freeway. *Transportmetrica B: Transp.* 1-24 (2022).
26. Chen, X., Li, L., & Zhang, Y. A Markov model for headway/spacing distribution of road traffic. *IEEE Trans. Intell. Transp. Syst.* **11**, 773-785 (2010).
27. Schreier, M., Willert, V., & Adamy, J. An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments. *IEEE Trans. Intell. Transp. Syst.* **17**, 2751-2766 (2016).
28. Schulz, J., Hubmann, C., Löchner, J., & Burschka, D. Interaction-aware probabilistic behavior prediction in urban environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 3999-4006 (IEEE, 2018).
29. Li, N. et al. Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems. *IEEE Trans. Control Syst. Technol*. **26**, 1782-1797 (2017).
30. Suo, S., Regalado, S., Casas, S., & Urtasun, R. TrafficSim: learning to simulate realistic multi-agent behaviors. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 10400-10409 (2021).
31. Kuefler, A., Morton, J., Wheeler, T., & Kochenderfer, M. Imitating driver behavior with generative adversarial networks. *IEEE Intelligent Vehicles Symposium (IV)* 204-211 (IEEE, 2017).
32. Bhattacharyya, R. et al. Multi-agent imitation learning for driving simulation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 1534-1539. (IEEE, 2018)
33. Bergamini, Luca. et al. SimNet: Learning reactive self-driving simulations from real-world observations. *IEEE International Conference on Robotics and Automation (ICRA)* 5119-5125 (IEEE, 2021).
34. Kamenev, A. et al. PredictionNet: real-time joint probabilistic traffic prediction for planning, control, and simulation. *International Conference on Robotics and Automation* (*ICRA*) 8936-8942. (IEEE, 2022).
35. Igl, M. et al. Symphony: Learning realistic and diverse agents for autonomous driving simulation. *International Conference on Robotics and Automation* (*ICRA*) 2445-2451. (IEEE, 2022).

36. Zhang, K., Chang, C., Zhong, W., Li, S., Li, Z., & Li, L. A Systematic Solution of Human Driving Behavior Modeling and Simulation for Automated Vehicle Studies. *IEEE Trans. Intell. Transp. Syst.* (2022).
37. Scanlon, J. M., Kusano, K. D., Daniel, T., Alderson, C., Ogle, A., & Victor, T. Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain. *Accid Anal & Prev*. **163**, 106454 (2021).
38. Liu, H. X., & Feng, S. "Curse of rarity" for autonomous vehicles. Preprint at https://arxiv.org/abs/2207.02749 (2022).
39. Bureau of Transportation Statistics. Transportation accidents by mode. https://www.bts.gov/content/transportation-accidents-mode.
40. Brown, T. et al. Language models are few-shot learners. *Advances in neural information processing systems*, **33**, 1877-1901 (2020).
41. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* **1**, 4171–4186 (2019).
42. Goodfellow, I. et al. Generative adversarial nets. *Advances in neural information processing systems*, **27**, (2014).
43. Ho, J., & Ermon, S. Generative adversarial imitation learning. *Advances in neural information processing systems*, **29**, (2016).
44. Allen, B. L., B. T. Shin, & P. J. Cooper. Analysis of Traffic Conflicts and Collisions. *Transport. Res. Rec.* **667**, 67–74 (1978).
45. What Michigan roundabouts were the most dangerous in 2020? https://www.michiganautolaw.com/blog/2021/08/18/michigan-roundabouts-most-dangerous-2020/.
46. Zou, Z., Zhang, R., Shen, S., Pandey, G., Chakravarty, P., Parchami, A., & Liu, H. X. Real-time Full-stack Traffic Scene Perception for Autonomous Driving with Roadside Cameras. *IEEE International Conference on Robotics and Automation (ICRA)* 890-896 (IEEE, 2022).
47. Zhang, R., Zou, Z., Shen, S., & Liu, H. X. Design, implementation, and evaluation of a roadside cooperative perception system. **2676**, 273-284 *Transp. Res. Rec.* (2022).
48. Krajewski, R., Moers, T., Bock, J., Vater, L., & Eckstein, L. The rounD dataset: a drone dataset of road user trajectories at roundabouts in Germany. *IEEE International Conference on Intelligent Transportation Systems (ITSC)* 1-6 (IEEE, 2020).
49. National Highway Traffic Safety Administration. 2020 FARS/CRSS coding and validation manual. Report No. DOT HS 813 251 (2022).
50. Michigan Traffic Crash Facts. https://www.michigantrafficcrashfacts.org/.
51. Meng, Y., Qin, Z., & Fan, C. Reactive and safe road user simulations using neural barrier certificates. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 6299-6306 (IEEE, 2021).

52. Sun, J., Zhang, H., Zhou, H., Yu, R., & Tian, Y. Scenario-based test automation for highly automated vehicles: A review and paving the way for systematic safety assurance. *IEEE Trans. Intell. Transp. Syst.* **23(9)**, 14088-14103 (2021).
53. Yu, H., Jiang, R., He, Z., Zheng, Z., Li, L., Liu, R., & Chen, X. Automated vehicle-involved traffic flow studies: A survey of assumptions, models, speculations, and perspectives. *Transp. Res. Part C Emerg*. **127**, 103101 (2021).
54. Vaswani, A. et al. Attention is all you need. *Advances in neural information processing systems*, **30**, (2017).
55. Dosovitskiy, A. et al. An image is worth 16x16 words: transformers for image recognition at scale. *International Conference on Learning Representations* (2021).
56. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, **596**, 583-589 (2021).
57. Ramesh, A. et al. Zero-shot text-to-image generation. *International Conference on Machine Learning,* 8821-8831 (2021).
58. Kitaev, N., Kaiser, Ł., & Levskaya, A. Reformer: The efficient transformer. *The International Conference on Learning Representations* (2020).
59. Choromanski, K. et al. Rethinking attention with performers. *The International Conference on Learning Representations* (2021).
60. Cui, H. et al. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. *International Conference on Robotics and Automation (ICRA).* 2090-2096 (IEEE, 2019).
61. Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. NeRF: representing scenes as neural radiance fields for view synthesis. *European conference on computer vision* 405-421 (Springer, Cham. 2020).
62. Shalev-Shwartz, S., Shammah, S., & Shashua, A. On a formal model of safe and scalable self-driving cars. Preprint at https://arxiv.org/abs/1708.06374 (2017).
63. Wang, J., Wu, J., & Li, Y. The driving safety field based on driver–vehicle–road interactions. *IEEE Trans. Intell. Transp. Syst.* **16**, 2203-2214 (2015).
64. Nistér, D., Lee, H. L., Ng, J., & Wang, Y. The safety force field. *NVIDIA whitepaper* (2019).
65. Pek, C., Manzinger, S., Koschi, M., & Althoff, M. Using online verification to prevent autonomous vehicles from causing accidents. Nat. Mach. Intell. 2, 518-528 (2020).
66. Yan, X., Zou, Z., Feng, S., Zhu, H., Sun, H., & Liu, H. X. Learning naturalistic driving environment with statistical realism. *Nat. Commun.* 14(1), 2037 (2023).