

# 1 Title

2 Learning Naturalistic Driving Environment with Statistical Realism

## 3 Authors

4 Xintao Yan<sup>1,6</sup>, Zhengxia Zou<sup>1,4,6</sup>, Shuo Feng<sup>1,2,5</sup>, Haojie Zhu<sup>1</sup>, Haowei Sun<sup>1</sup>, Henry X. Liu<sup>1,2,3,\*</sup>

## 5 Affiliations

6 <sup>1</sup>Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI, USA

7 <sup>2</sup>University of Michigan Transportation Research Institute, Ann Arbor, MI, USA

8 <sup>3</sup>Mcity, University of Michigan, Ann Arbor, MI, USA

9 <sup>4</sup>Present Address: School of Astronautics, Beihang University, Beijing, China

10 <sup>5</sup>Present Address: Department of Automation, Tsinghua University, Beijing, China

11 <sup>6</sup>These authors contributed equally: Xintao Yan, Zhengxia Zou

12 \*Corresponding Author, henryliu@umich.edu

## 13 Abstract

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

## 25 Introduction

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

34 In recent years, great efforts have been made in developing simulators for autonomous driving systems. Thanks  
35 to rapid advances in artificial intelligence (AI), computer vision and graphics, and high-performance computing  
36 devices, accurate vehicle dynamics, photorealistic rendering, and realistic sensor simulation are now being  
37 realized and accessible. Some well-known simulators include Intel's CARLA<sup>8</sup>, Google/Waymo's CarCraft<sup>9</sup> and  
38 SimulationCity<sup>5</sup>, Tesla's simulator<sup>10</sup>, Microsoft's AirSim<sup>11</sup>, NVIDIA's DRIVE Sim<sup>12</sup>, Baidu's AADS<sup>3</sup>, and  
39 Cruise's simulator<sup>13</sup>, etc. Despite the above efforts and advancements, these simulators mainly focus on the

40 fidelity of the vehicle rather than the driving environment, especially for the background road user behavior. The  
41 behaviors of background agents are either replayed from logged data or simulated using oversimplified heuristic  
42 rules, which leaves a significant gap between the simulation and the real-world driving environment.

43 The key to high-fidelity NDE is accurate modeling of human driving behavior. Microscopic traffic simulators,  
44 which mimic the interactive agent behaviors through a combination of physics-driven models and hand-crafted  
45 rules, such as car-following models<sup>14,15</sup>, lane-changing models<sup>16,17</sup>, gap-acceptance models<sup>18</sup>, etc., have been  
46 studied and developed in the transportation engineering domain for decades. Some well-known traffic simulators  
47 are SUMO<sup>19</sup>, VISSIM<sup>20</sup>, and AIMSUN<sup>21</sup>. Due to the limited capability of the underlying parametric models and  
48 manually encoded rules, the model fidelity is constrained. Many attempts have been made by using neural  
49 networks<sup>22-27</sup>, Markovian-based models<sup>6,28</sup>, Bayesian networks<sup>29,30</sup>, and game theory<sup>31</sup>, etc., to achieve better  
50 performance in modeling specific behaviors (e.g., car-following) or specific scenarios (e.g., unprotected left turn).  
51 However, they can hardly be generalized and scaled to model complex urban environments and highly interactive  
52 scenarios.

53 The focus of this study is to build a high-fidelity simulator that is statistically representative of real-world driving  
54 environments, particularly for those long-tail safety-critical events. Especially, we aim to produce safety-critical  
55 events with distribution-level accuracy, including both crashes and near-misses, which are critical for training  
56 and testing AVs. This differentiates our proposed NeuralNDE model from most existing simulators based on  
57 imitation learning (including generative adversarial imitation learning)<sup>32-38</sup>, where statistical realism is hardly  
58 considered and cannot be achieved. For example, the crash rates of these simulation environments are  
59 significantly higher (e.g., SimNet<sup>35</sup>) than that of real-world traffic. Moreover, these methods can only generate  
60 short-time simulations in the order of a few seconds (e.g., D2Sim<sup>38</sup>), which limits the capability of full-length trip  
61 training and evaluation of AVs. To reproduce high-fidelity safety-critical events, there are also methods proposed  
62 based on real-world event reconstruction. For example, the researchers constructed the simulation environment  
63 based on real-world fatal collision events from various data sources including police reports<sup>39</sup>. However, it may  
64 be difficult to reconstruct near-miss events using this method since the information needed for reconstruction is  
65 usually not available. Therefore, these reconstruction-based methods and our learning-based method serve to  
66 complement each other when building high-fidelity simulators.

67 The lack of statistical realism for simulation can potentially mislead AV development in both training and testing.  
68 An illustration example is shown in Fig. 1a. Consider a roundabout environment that includes multiple vehicles.  
69 At time  $t$ , a vehicle (vehicle 1) is circulating, and another vehicle (vehicle 2) is about to enter the roundabout.  
70 Their potential future positions are denoted by shaded blue areas, and they have a probability to collide if vehicle  
71 2 fails to yield. Assume the distance between the two vehicles in the real world follows certain distribution as  
72 shown by the red curve and the simulated results are the dashed blue curve. This statistical difference, i.e.,  
73 distribution inconsistency between the real world and simulation, will lead to an underestimation of vehicle crash  
74 rate and therefore provide optimistic estimates of AV safety performance. Also, since the distance between  
75 vehicles in the simulation environment is not consistent with the real world, an AV agent trained in it might not  
76 fit in real traffic due to the large sim-to-real gap. In real-world driving environments, instead of two agents,  
77 multiple human drivers are continuously interacting with each other and their states are progressively evolving  
78 for a long time horizon. Therefore, the underlying joint NDE distribution is extremely complex and in a very  
79 high-dimensional space as shown in Fig. 1b. The goal of NDE modeling is to achieve distribution-level accuracy  
80 under both normal driving and safety-critical situations. Therefore, a wide range of environment statistics, for  
81 example, vehicle speed and distance distributions, crash rate, crash type and severity distributions, near-miss  
82 measurements, etc., need to be consistent with the real world.

83 The challenges of modeling NDE with statistical realism mainly come from three aspects as shown in Fig. 1c.  
84 The first challenge is from the “curse of dimensionality”. The real-world driving environment is highly interactive

85 and spatiotemporally complex with large numbers of road users and long time horizons, which make NDE  
86 modeling a very high-dimensional problem. The second challenge is from the “curse of rarity”<sup>40</sup>. Since safety-  
87 critical events (e.g., crashes) rarely happen in the real-world driving environment (on average  $10^{-6}$  crashes per  
88 driving mile for human drivers<sup>41</sup>), modeling such rare events in high fidelity requires an extremely high precision  
89 of the microscopic behavior. The compounding effects of the “curse of rarity” on top of the “curse of  
90 dimensionality” in the real world NDE will make it even more challenging<sup>40</sup>. The third challenge is from the  
91 “distribution shift”<sup>32</sup>, which is particularly critical for learning-based simulators. Short-term and small modeling  
92 errors may accumulate both in space and time, which might lead to out-of-distribution behaviors like frequent  
93 offroad, unrealistic collision, or even the collapse of the entire simulation. Moreover, due to the highly interactive  
94 nature of the driving environment, unrealistic behaviors of a single agent will impact and propagate to all agents  
95 in the simulation.

96 In this paper, we solve this long-standing problem by developing NeuralNDE - a novel deep learning-based  
97 framework for simulating Naturalistic Driving Environment with statistical realism. The overview of the proposed  
98 framework is shown in Fig. 2a. We frame the simulation modeling under an imitation learning paradigm with  
99 deep neural networks under the supervision of large-scale real-world demonstration. The behavior modeling  
100 network takes in all road users' past states within a historical time window as input and predicts their joint  
101 distribution of future actions. We leverage the recent advances in fundamental models (e.g., GTP<sup>42</sup> and BERT<sup>43</sup>)  
102 and use Transformer as the backbone of the behavior modeling network to characterize multi-agent interaction  
103 behaviors. The behavior modeling network can achieve distribution-level accuracy in normal driving conditions,  
104 however, it cannot achieve such accuracy in safety-critical conditions, due to the rarity of safety-critical events in  
105 the training data, which will lead to inaccurate statistics like unrealistically high crash rates. To tackle this issue,  
106 a conflict critic mechanism is introduced during the inference time as shown in Fig. 2b. It will monitor the  
107 generated trajectories, and if there is a potential conflict, there is a certain probability to accept vehicles  
108 performing dangerous behavior, which makes NeuralNDE capable of realizing accurate safety-critical statistics.  
109 Otherwise, the generated behaviors will be guided and rectified by the safety mapping network to resolve the  
110 conflict. The differentiable safety mapping network is a neural mapper pretrained from physics and driving rules  
111 to map unsafe behaviors to a feasible domain of safety. To further overcome the distribution shift issue, we  
112 integrate the generative adversarial training as in GAN<sup>44</sup> and GAIL<sup>45</sup>, where a discriminator is introduced to be  
113 jointly trained with the behavior modeling network. During the simulation process, as shown in Fig. 2c, the state  
114 of all road users will be updated based on the behavior modeling network, conflict critic module, and the safety  
115 mapping network in each simulation step to autoregressively generate the simulation environment.

116 To demonstrate the effectiveness of our approach, we construct two multi-lane roundabout environments located  
117 in the US and Germany, respectively, using real-world data. An illustration video is provided in Supplementary  
118 Movie 1. The simulated environment is validated to be statistically accurate with the real world, including vehicle  
119 instantaneous speed, distance, and yielding behavior. More importantly, the proposed NeuralNDE can achieve  
120 accurate safety-critical statistics including both crash and near-miss measurements, for example, crash rate, crash  
121 type, crash severity, post-encroachment time (PET<sup>46</sup>), etc. The fidelity of NeuralNDE-generated crash events is  
122 further validated against real-world crash videos and police crash reports. To the best of our knowledge, this is  
123 the first time that a simulation environment can systematically reproduce the real-world driving environment with  
124 statistical realism, particularly for those long-tail safety-critical events that are critical to AV safety. In addition,  
125 the proposed environment can perform long-time (hour-level) simulation, where the AV under training or testing  
126 can continuously interact with background vehicles. The proposed NeuralNDE should be readily integrated with  
127 different high-fidelity AV simulators, for example, CARLA<sup>5</sup>, which focuses on photorealistic rendering and  
128 sensor simulations, to provide a realistic traffic environment. Furthermore, it should be noted that the proposed  
129 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.

## Results

### Dataset

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 Fig. 3a. This is a busy roundabout with a large traffic volume and the fourth-highest crash rate in Michigan<sup>47</sup>. A roadside perception system<sup>48,49</sup> 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 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, real-world safety-critical rare-event data are not available in most existing public datasets, however, they are essential for constructing and validating the performance of generated simulation environments. More details about the dataset can be found in Supplementary Materials.

We also use an open dataset, round<sup>50</sup>, which is collected at a two-lane roundabout located at Neuweiler, Aachen, Germany, to further demonstrate NeuralNDE performance in normal driving conditions. Due to space limits, we will use results on AA datasets to illustrate the performance, and the results using the round dataset can be found in Supplementary Materials.

### 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. More details about experiment settings can be found in Supplementary Materials.

### 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. Detailed definitions for the metrics are introduced in the following paragraphs.

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 Supplementary Fig. 1a. 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 study.

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 Fig. 3b. Vehicles in the corresponding circle quadrant as shown in Fig. 3b 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.

Two agents are considered in a crash if their rectangle 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<sup>51</sup>. More information can be found in Supplementary Materials.

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 information can be found in Supplementary Materials.

The post-encroachment time (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  $1.3 \times 1.3$  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:

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

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

$$D_{KL}(\mathbf{P}, \mathbf{Q}) = \sum_x P(x) \log \frac{P(x)}{Q(x)}. \quad (2)$$

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

We compare the proposed method with SUMO<sup>19</sup> - a widely used simulation platform for traffic environments, and other state-of-the-art methods. More details for the SUMO simulator settings can be found in Supplementary Materials.

### 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.

210 An illustration video is provided in Supplementary Movie 2. Vehicle speed and position are direct outcomes of  
211 microscopic driving behaviors, and they are critical for both training and testing the AV. The proposed  
212 NeuralNDE can generate accurate vehicle instantaneous speed distribution as in the real world, as shown in Fig.  
213 4a. Compared with the SUMO baseline, vehicle speeds in NeuralNDE are naturally distributed among the whole  
214 range, covering both low and high-speed situations. Furthermore, NeuralNDE can also accurately reproduce  
215 vehicle distance distribution as shown in Fig. 4b, reflecting the full distribution of encounters that AV might face  
216 in the real world.

217 In a two-lane roundabout environment, a highly interactive location is at the roundabout entrance where entering  
218 vehicles need to yield to conflicting vehicles within the roundabout. Many real-world conflicts and crashes occur  
219 in this location, and the fidelity of these safety-critical events depends on the accuracy of the yielding behavior.  
220 Therefore, we will examine the yielding behavior simulated by NeuralNDE to further demonstrate its fidelity in  
221 modeling human interactions. The yielding behavior depends on the distance to the conflicting vehicle and the  
222 speed of the conflicting vehicle that is traveling within the roundabout. The results of yielding distance and  
223 yielding speed distributions are shown in Fig. 4c and Fig. 4d, respectively. We can find that NeuralNDE can  
224 accurately replicate human yielding behavior and significantly outperforms the SUMO simulator. Human drivers  
225 are naturally heterogeneous and have different characteristics. Different drivers often exhibit diverse driving  
226 behaviors and make different decisions, for example, some drivers are more aggressive and only give way when  
227 conflicting vehicles are very close while others might be more conservative. The proposed NeuralNDE is directly  
228 learned from real-world data without hand-crafted rules, therefore, it can master the nuanced yielding behavior  
229 of human drivers and generate a realistic and diverse driving environment.

### 230 **Statistical realism of safety-critical driving behavior**

231 The key challenge of current AV development is how to handle safety-critical driving situations occurring in the  
232 real world, therefore, the simulation environment must be able to reproduce these long-tail rare events with high  
233 fidelity. In this section, we will examine the performance of NeuralNDE in generating safety-critical events,  
234 which include both crash and near-miss situations. The first important statistic is the crash rate. The ground-truth  
235 crash rate of the studied roundabout is  $1.21 \times 10^{-4}$  crash/km (more details can be found in Supplementary  
236 Materials). The crash rate of the NeuralNDE is  $1.25 \times 10^{-4}$  crash/km, which can accurately reproduce the real-  
237 world ground truth.

238 Not only can the proposed NeuralNDE reproduce an accurate crash rate, but also the detailed composition of  
239 crash types and crash severity distribution as shown in Fig. 5a and Fig. 5b, respectively. The ground-truth crash  
240 type and crash severity distributions are summarized from all crash events that happened at this roundabout from  
241 2016 to 2020 based on police crash reports<sup>52</sup> (more details can be found in Supplementary Materials). These  
242 demonstrate that NeuralNDE can generate accurate and diverse crash events following real-world occurring  
243 patterns, which are crucial for comprehensive testing of AV performance in different potential crashes. Compared  
244 with most state-of-the-art methods, for example, Refs<sup>32,35,36,37,53</sup>, none of them compared their simulation results  
245 (e.g., crash rates/types/severities) against the real-world data. To the best of our knowledge, we are the only study  
246 that validated the simulated safety-critical statistics with real-world ground truth. For each crash type, we will  
247 further compare NeuralNDE-generated and real-world crash events in the later section to qualitatively  
248 demonstrate the fidelity of our approach. These results validate the capability and effectiveness of the proposed  
249 NeuralNDE in generating accurate crash statistics, which is critical for AV applications.

250 In addition to crashes, near-miss situations are also important. Two measurements, vehicle distance and PET  
251 distributions, are examined to validate the NeuralNDE fidelity. The closest distance between vehicles objectively  
252 characterizes potential conflicts between them. To validate the near-miss fidelity, we will focus on the vehicle  
253 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 Fig. 5c. 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 Fig. 5d. 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.

To validate the effectiveness of each module of the proposed framework, we conducted ablation studies and the result can be found in Supplementary Materials. It demonstrates that the Transformer-based behavior modeling network and the generative adversarial training technique are important for modeling multi-agent interaction behavior, and conflict critic module and safety mapping network are essential for generating realistic safety-critical events.

## 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, as shown in Supplementary Movie 1. 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 Fig. 6. The video version of these events can be found in Supplementary Movie 3.

The first case is an angle crash caused by failure to yield as shown in Fig. 6a, 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 Fig. 6a. 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 Fig. 6b. 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 Fig. 6b, 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 Fig. 6c. 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.

## 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 Fig. 7, that involves two scenarios, i.e., a four-way stop sign-controlled intersection and a two-lane roundabout. We use SUMO<sup>19</sup> simulator to generate vehicle trajectory data and use it as the ground truth of the NDE. We assume the traffic network is not fully perceptual 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<sup>15</sup> and SL2015 lane-changing model<sup>19</sup>. The proposed method can generate high-fidelity simulation, as shown in Supplementary Movie 4. More details about experiment settings can be found in Supplementary Materials.

We simulate the network and collect the data in intersection and roundabout areas to quantitatively 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 Fig. 8a-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 Fig. 8c-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 Fig. 9a-d, respectively. The safety-critical events metrics (vehicle distance in the near-miss situations and PET) are shown in Fig. 9e-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.

## Discussion

The proposed NeuralNDE demonstrates promising performance 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.

For simulation-based AV testing, there are two critical problems that need to be solved. The first one is how to build a high-fidelity simulation environment and the second one is how to develop an accelerated testing methodology that can evaluate the AV performance accurately and efficiently. This work is focusing on the first problem, where the high-fidelity simulator is a prerequisite and foundation for simulation-based AV testing applications<sup>2,4,54</sup>. In our prior works<sup>2,4</sup>, we have developed accelerated testing methodologies that use a purely

data-driven NDE model, which can be replaced with the high-fidelity NeuralNDE. We should note that human-driven vehicles might behave differently if they are interacting with an AV<sup>55</sup>. Therefore, further development and enhancement for NeuralNDE might be required in the future to consider the AV influences on surrounding vehicles.

## Methods

### 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  $\tau$  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<sup>56</sup>, and have revealed remarkable performance in many applications, including computer vision<sup>57</sup>, bioinformatics<sup>58</sup>, and multimodal data generative modeling<sup>59</sup>.

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<sup>60,61</sup> 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  $F_M$  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:

$$p(\mathbf{S}_N^{\tau:t}) = F_M(\mathbf{S}_N^{\tau:t}) \sim N(\boldsymbol{\mu}_{\mathbf{a},i=1\dots N}^t, \boldsymbol{\Sigma}_{\mathbf{a},i=1\dots N}^t), \quad (3)$$

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

$$\mathbf{a}_1^t, \dots, \mathbf{a}_N^t \leftarrow N(\boldsymbol{\mu}_{\mathbf{a},i=1\dots N}^t, \boldsymbol{\Sigma}_{\mathbf{a},i=1\dots N}^t). \quad (4)$$

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:

$$\mathbf{s}_i^{t+1} = T(\mathbf{a}_i^t, \mathbf{s}_i^t). \quad (5)$$

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.,  $\kappa$  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<sup>62</sup>.

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 Fig. 10a. 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<sup>43</sup> 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.<sup>63</sup> 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  $\gamma$  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:

$$\gamma(\mathbf{s}) = [\mathbf{s}, \sin(2^0\pi\mathbf{s}), \cos(2^0\pi\mathbf{s}), \dots, \sin(2^{L-1}\pi\mathbf{s}), \cos(2^{L-1}\pi\mathbf{s})]. \quad (6)$$

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  $\boldsymbol{\mu}_{\mathbf{a},i}^t$ , another for predicting  $\boldsymbol{\Sigma}_{\mathbf{a},i}^t$ . 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:

$$p(F_M) = p(\mathbf{s}_1^1, \mathbf{s}_1^2, \dots, \mathbf{s}_1^t, \dots, \mathbf{s}_N^T). \quad (7)$$

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

$$p(\mathbf{s}_i^t) = \prod_{j=1}^D \frac{1}{(2\pi)^{\frac{1}{2}}} \frac{1}{\sigma_{i,j}^t} \exp\left\{-\frac{(\mu_{i,j}^t - a_{i,j}^t)^2}{2(\sigma_{i,j}^t)^2}\right\}, \quad (8)$$

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

We approximate the joint probability distribution  $p(F_M)$  in Equation (7) as the multiplicative form of each agent's marginal probabilities, and combine it with Equation (8) to derive the loss function in the negative log-likelihood form as follows:

420

$$L_M(F_M) = \sum_{t=1}^T \sum_{i=1}^N \sum_{j=1}^D \left[ \ln(\sigma_{i,j}^t) + \frac{(\mu_{i,j}^t - a_{i,j}^t)^2}{2(\sigma_{i,j}^t)^2} \right]. \quad (9)$$

421

422

423

424

425

426

The approximation of Equation (7) 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 (9), 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. The training details can be found in Supplementary Materials.

427

### Conflict critic module

428

429

430

431

432

433

434

435

436

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 Fig. 2b. The input to  $F_c$  are the sampled  $\kappa$  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  $F_c$  is the acceptance probability  $p_a$  for not passing through the safety mapping network:

437

$$p_a = F_c(\mathbf{S}_N^{t:\kappa}). \quad (10)$$

438

439

440

441

442

443

444

445

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  $F_c$  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.

446

447

448

449

450

451

452

453

454

In this study, 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 Fig. 10b. 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.

455

456

457

458

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

459 the current NeuralNDE crash rate at the first iteration  $c^1$ . Then linearly update the uniform acceptance probability  
 460 as follows

$$461 \quad p_{ua}^{i+1} = c^{gt} \cdot \frac{p_{ua}^i}{c^i}, \quad (11)$$

462 where  $c^{gt}$  denotes the desired ground-truth crash rate, and  $i$  denotes the current iteration number. Continue this  
 463 process until the NeuralNDE crash rate is close to the ground truth with satisfactory accuracy. In the second step,  
 464 we will calibrate the acceptance probability for each crash type. The acceptance probability  $p_a(j)$  for crash type  
 465  $j$  needs to satisfy the following system of linear equations to fit both crash rate (Equation (12)) and crash type  
 466 distribution (Equation (13)):

$$467 \quad \sum_j p(j)p_a(j) = p_{ua}. \quad (12)$$

$$468 \quad \frac{p(j)p_a(j)}{\sum_j p(j)p_a(j)} = c^{gt}(j), \forall j \in J, \quad (13)$$

469 where  $c^{gt}(j)$  is the ground-truth probability of crash type  $j$ ,  $p_{ua}$  is the uniform acceptance probability obtained  
 470 from the first step, and  $p(j)$  is the probability of crash type  $j$  occurring in NeuralNDE using the uniform  
 471 probability  $p_{ua}$ . For Equation (12), the summation of  $p(j)p_a(j)$  over all potential crash types  $j$  is the overall  
 472 acceptance probability considering different crash types. It needs to be equal to the uniform acceptance probability  
 473 ( $p_{ua}$ ) obtained in the first step, which can guarantee the accurate crash rate of the simulation. Therefore, the  
 474 acceptance probability  $p_a(j)$  equals to

$$475 \quad p_a(j) = p_{ua} \cdot \frac{c^{gt}(j)}{p(j)}. \quad (14)$$

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

## 477 Safety mapping network

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

$$483 \quad \mathbf{A}_N^{t:\kappa,*} = F_S(\mathbf{S}_N^t, \mathbf{A}_N^{t:\kappa}). \quad (15)$$

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

488 The safety mapping network is trained to imitate existing model-based safety guards based on domain knowledge.  
 489 In this study, for simplicity and generality, we consider one-step prediction and use a physics-based safety guard  
 490 as the training target. The illustration figure is shown in Supplementary Fig. 2. When two vehicles are going to  
 491 collide with each other, we resolve the potential conflict by setting a repulsive force between them. The force is  
 492 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:

$$L_S(F_S) = \|F_S(\mathbf{S}_N^t, \mathbf{A}_N^t) - \widehat{\mathbf{A}}_N^t\|_1, \quad (16)$$

where  $\widehat{\mathbf{A}}_N^t$  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 Fig. 2a.

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<sup>64</sup>, potential force field-based methods<sup>65,66</sup>, online verification methods<sup>67</sup>, etc.

### 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<sup>44</sup>. 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,  $\widehat{\mathbf{S}} \sim p_R(\mathbf{S}_N^t)$  represents a trajectory sampled from the real-world data distribution, and  $\mathbf{S} \sim p_G(\mathbf{S}_N^t)$  represents a trajectory generated from the simulation. We follow a standard adversarial training pipeline and define the adversarial loss functions as follows:

$$L_{\text{adv}}(F_M, D) = E_{\widehat{\mathbf{S}} \sim p_R(\mathbf{S}_N^t)}[\log D(\widehat{\mathbf{S}})] + E_{\mathbf{S} \sim p_G(\mathbf{S}_N^t)}[\log(1 - D(\mathbf{S}))]. \quad (17)$$

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:

$$F_M^*, D^* = \arg \min_{F_M} \max_D \{L_M(F_M) + \beta L_{\text{adv}}(F_M, D)\}. \quad (18)$$

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

## 532 Network architecture

533 The network architecture of the behavior modeling network is shown in Fig. 10a. Each road user is considered as  
534 a token and the input to the network is its historical trajectory, i.e., position (x, y coordinates) and heading (cosine  
535 and sine of heading), of all road users within the historical time window. The number of historical steps at the  
536 input is set to  $\tau = 5$  with a resolution of 0.4 seconds per step. Then, the input will pass through the frequency  
537 encoding<sup>63</sup> layer, in which we apply a set of sine and cosine basis functions that project the vectors to high  
538 dimensional space to improve capturing high-frequency variation in the state spaces. Suppose  $\gamma$  defines a  
539 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  
540 embedding layer is a fully connected layer that converts the state dimension to the Transformer hidden layer  
541 dimension. Then a set of ( $N = 4$ ) standard BERT<sup>43</sup> transformer layers is stacked together. The dimension of the  
542 hidden layer in the Transformer block is 256, the number of heads in multi-headed attention layers is 4, the  
543 dimension of intermediate layers in the position-wise feedforward net is 512, the probability of dropout of various  
544 hidden layers is 0, and the probability of dropout of attention layers is 0. Then, the output from the Transformer  
545 will pass through prediction heads (single layer MLP with 256 neurons) to generate the final output. In practice,  
546 we directly predict the states of each vehicle rather than actions for simplicity. Therefore, the output of the  
547 behavior modeling network is the predicted trajectory, i.e., stochastic position (one prediction head for the mean  
548 of position, one prediction head for the variance of position, and one prediction head for deterministic heading),  
549 of each vehicle in the prediction time horizon. The number of prediction steps at the output is set to  $\kappa = 5$  with a  
550 resolution of 0.4 seconds per step.

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

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

## 559 Data availability

560 The Ann Arbor roundabout dataset that we used to train NeuralNDE is publicly available at  
561 <https://github.com/michigan-traffic-lab/Learning-Naturalistic-Driving-Environment>. The ground-truth crash type,  
562 crash severity distributions, and police crash reports are available at <https://www.michigantrafficcrashfacts.org/>.  
563 The background image of the simulated Ann Arbor environment is from Google Maps. The round dataset that  
564 we used to train NeuralNDE is publicly available at <https://www.round-dataset.com/> for non-commercial usage.  
565 The background image of the simulated round environment is also from the round dataset. Source data for figures  
566 are provided in the Source Data file.

## 567 Code availability

568 The SUMO simulation platform is publicly available at <https://www.eclipse.org/sumo/>. The source code used for  
569 the generation of NeuralNDE is available from the corresponding author upon reasonable request.

## 570 References

571 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*. In press.
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. *Atlantic* **23**, 3-1 (2017).
10. Tesla. Tesla AI Day. <https://www.youtube.com/watch?v=j0z4FweCy4M> (2021).
11. 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).
12. NVIDIA. NVIDIA DRIVE Sim. <https://developer.nvidia.com/drive/drive-sim> (2021).
13. Cruise. Cruise autonomous vehicles "drive" 200,000 Hours a day in simulation. [https://www.youtube.com/watch?v=9G2zUUTeG1I&ab\\_channel=Cruise](https://www.youtube.com/watch?v=9G2zUUTeG1I&ab_channel=Cruise) (2020).
14. Bando, M., Hasebe, K., Nakayama, A., Shibata, A., & Sugiyama, Y. Dynamical model of traffic congestion and numerical simulation. *Phys. Rev. E.* **51**, 1035 (1995).
15. Treiber, M., Hennecke, A., & Helbing, D. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E.* **62**, 1805 (2000).
16. Kesting, A., Treiber, M., & Helbing, D. General lane-changing model MOBIL for car-following models. *Transport. Res. Rec.* **1999**, 86-94 (2007).
17. Erdmann, J. SUMO's lane-changing model. In *Modeling Mobility with Open Data*. (Springer, Cham 2015).
18. Mahmassani, H., & Sheffi, Y. Using gap sequences to estimate gap acceptance functions. *Transp. Res. B. Methodol.* **15**, 143-148 (1981).
19. Lopez, P. et al. Microscopic traffic simulation using SUMO. *International Conference on Intelligent Transportation Systems (ITSC)* 2575-2582 (IEEE, 2018).
20. VISSIM. Vissim 5.40-01, user manual. Planung Transport Verkehr AG, Karlsruhe, Germany (2012).
21. AIMSUN. AIMSUN Next. <https://www.aimsun.com/aimsun-next> (2022).
22. Wang, X. et al. Capturing car-following behaviors by deep learning. *IEEE Trans. Intell. Transp. Syst.* **19**, 910-920 (2017).
23. 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).
24. 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).
25. Mo, Z., Shi, R., & Di, X. A physics-informed deep learning paradigm for car-following models. *Transp. Res. C. Emerg. Technol.* **130**, 103240 (2021).
26. 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).
27. 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).

- 618 28. Chen, X., Li, L., & Zhang, Y. A Markov model for headway/spacing distribution of road traffic. *IEEE*  
619 *Trans. Intell. Transp. Syst.* **11**, 773-785 (2010).
- 620 29. Schreier, M., Willert, V., & Adamy, J. An integrated approach to maneuver-based trajectory prediction and  
621 criticality assessment in arbitrary road environments. *IEEE Trans. Intell. Transp. Syst.* **17**, 2751-2766  
622 (2016).
- 623 30. Schulz, J., Hubmann, C., Löchner, J., & Burschka, D. Interaction-aware probabilistic behavior prediction in  
624 urban environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 3999-  
625 4006 (IEEE, 2018).
- 626 31. Li, N. et al. Game theoretic modeling of driver and vehicle interactions for verification and validation of  
627 autonomous vehicle control systems. *IEEE Trans. Control Syst. Technol.* **26**, 1782-1797 (2017).
- 628 32. Suo, S., Regalado, S., Casas, S., & Urtasun, R. TrafficSim: learning to simulate realistic multi-agent  
629 behaviors. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 10400-  
630 10409 (2021).
- 631 33. Kuefler, A., Morton, J., Wheeler, T., & Kochenderfer, M. Imitating driver behavior with generative  
632 adversarial networks. *IEEE Intelligent Vehicles Symposium (IV)* 204-211 (IEEE, 2017).
- 633 34. Bhattacharyya, R. et al. Multi-agent imitation learning for driving simulation. *IEEE/RSJ International*  
634 *Conference on Intelligent Robots and Systems (IROS)* 1534-1539. (IEEE, 2018)
- 635 35. Bergamini, Luca. et al. SimNet: Learning reactive self-driving simulations from real-world observations.  
636 *IEEE International Conference on Robotics and Automation (ICRA)* 5119-5125 (IEEE, 2021).
- 637 36. Kamenev, A. et al. PredictionNet: real-time joint probabilistic traffic prediction for planning, control, and  
638 simulation. *International Conference on Robotics and Automation (ICRA)* 8936-8942. (IEEE, 2022).
- 639 37. Igl, M. et al. Symphony: Learning realistic and diverse agents for autonomous driving simulation.  
640 *International Conference on Robotics and Automation (ICRA)* 2445-2451. (IEEE, 2022).
- 641 38. Zhang, K., Chang, C., Zhong, W., Li, S., Li, Z., & Li, L. A Systematic Solution of Human Driving Behavior  
642 Modeling and Simulation for Automated Vehicle Studies. *IEEE Trans. Intell. Transp. Syst.* (2022).
- 643 39. Scanlon, J. M., Kusano, K. D., Daniel, T., Alderson, C., Ogle, A., & Victor, T. Waymo simulated driving  
644 behavior in reconstructed fatal crashes within an autonomous vehicle operating domain. *Accid Anal & Prev.*  
645 **163**, 106454 (2021).
- 646 40. Liu, H. X., & Feng, S. "Curse of rarity" for autonomous vehicles. Preprint at  
647 <https://arxiv.org/abs/2207.02749> (2022).
- 648 41. Bureau of Transportation Statistics. Transportation accidents by mode.  
649 <https://www.bts.gov/content/transportation-accidents-mode>.
- 650 42. Brown, T. et al. Language models are few-shot learners. *Advances in neural information processing*  
651 *systems*, **33**, 1877-1901 (2020).
- 652 43. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers  
653 for language understanding. *Conference of the North American Chapter of the Association for*  
654 *Computational Linguistics: Human Language Technologies*, **1**, 4171-4186 (2019).
- 655 44. Goodfellow, I. et al. Generative adversarial nets. *Advances in neural information processing systems*, **27**,  
656 (2014).
- 657 45. Ho, J., & Ermon, S. Generative adversarial imitation learning. *Advances in neural information processing*  
658 *systems*, **29**, (2016).
- 659 46. Allen, B. L., B. T. Shin, & P. J. Cooper. Analysis of Traffic Conflicts and Collisions. *Transport. Res. Rec.*  
660 **667**, 67-74 (1978).
- 661 47. What Michigan roundabouts were the most dangerous in 2020?  
662 <https://www.michiganautolaw.com/blog/2021/08/18/michigan-roundabouts-most-dangerous-2020/>.

- 663 48. Zou, Z., Zhang, R., Shen, S., Pandey, G., Chakravarty, P., Parchami, A., & Liu, H. X. Real-time Full-stack  
664 Traffic Scene Perception for Autonomous Driving with Roadside Cameras. *IEEE International Conference*  
665 *on Robotics and Automation (ICRA)* 890-896 (IEEE, 2022).
- 666 49. Zhang, R., Zou, Z., Shen, S., & Liu, H. X. Design, implementation, and evaluation of a roadside cooperative  
667 perception system. **2676**, 273-284 *Transp. Res. Rec.* (2022).
- 668 50. Krajewski, R., Moers, T., Bock, J., Vater, L., & Eckstein, L. The round dataset: a drone dataset of road user  
669 trajectories at roundabouts in Germany. *IEEE International Conference on Intelligent Transportation*  
670 *Systems (ITSC)* 1-6 (IEEE, 2020).
- 671 51. National Highway Traffic Safety Administration. 2020 FARS/CRSS coding and validation manual. Report  
672 No. DOT HS 813 251 (2022).
- 673 52. Michigan Traffic Crash Facts. <https://www.michigantrafficcrashfacts.org/>.
- 674 53. Meng, Y., Qin, Z., & Fan, C. Reactive and safe road user simulations using neural barrier certificates.  
675 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 6299-6306 (IEEE, 2021).
- 676 54. Sun, J., Zhang, H., Zhou, H., Yu, R., & Tian, Y. Scenario-based test automation for highly automated  
677 vehicles: A review and paving the way for systematic safety assurance. *IEEE Trans. Intell. Transp. Syst.*  
678 **23(9)**, 14088-14103 (2021).
- 679 55. Yu, H., Jiang, R., He, Z., Zheng, Z., Li, L., Liu, R., & Chen, X. Automated vehicle-involved traffic flow  
680 studies: A survey of assumptions, models, speculations, and perspectives. *Transp. Res. Part C Emerg.* **127**,  
681 103101 (2021).
- 682 56. Vaswani, A. et al. Attention is all you need. *Advances in neural information processing systems*, **30**, (2017).
- 683 57. Dosovitskiy, A. et al. An image is worth 16x16 words: transformers for image recognition at scale.  
684 *International Conference on Learning Representations* (2021).
- 685 58. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, **596**, 583-589 (2021).
- 686 59. Ramesh, A. et al. Zero-shot text-to-image generation. *International Conference on Machine Learning*,  
687 8821-8831 (2021).
- 688 60. Kitaev, N., Kaiser, Ł., & Levskaya, A. Reformer: The efficient transformer. *The International Conference*  
689 *on Learning Representations* (2020).
- 690 61. Choromanski, K. et al. Rethinking attention with performers. *The International Conference on Learning*  
691 *Representations* (2021).
- 692 62. Cui, H. et al. Multimodal trajectory predictions for autonomous driving using deep convolutional networks.  
693 *International Conference on Robotics and Automation (ICRA)*. 2090-2096 (IEEE, 2019).
- 694 63. Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. NeRF: representing  
695 scenes as neural radiance fields for view synthesis. *European conference on computer vision* 405-421  
696 (Springer, Cham. 2020).
- 697 64. Shalev-Shwartz, S., Shammah, S., & Shashua, A. On a formal model of safe and scalable self-driving cars.  
698 Preprint at <https://arxiv.org/abs/1708.06374> (2017).
- 699 65. Wang, J., Wu, J., & Li, Y. The driving safety field based on driver-vehicle-road interactions. *IEEE Trans.*  
700 *Intell. Transp. Syst.* **16**, 2203-2214 (2015).
- 701 66. Nistér, D., Lee, H. L., Ng, J., & Wang, Y. The safety force field. *NVIDIA whitepaper* (2019).
- 702 67. Pek, C., Manzinger, S., Koschi, M., & Althoff, M. Using online verification to prevent autonomous vehicles  
703 from causing accidents. *Nat. Mach. Intell.* **2**, 518-528 (2020).

## 704 Acknowledgments

705 This research was partially funded by the U.S. Department of Transportation (USDOT) Region 5 University  
706 Transportation Center: Center for Connected and Automated Transportation (CCAT) of the University of  
707 Michigan (69A3551747105) and the National Science Foundation (CMMI #2223517). Any opinions, findings,

708 and conclusions or recommendations expressed in this material are those of the authors and do not necessarily  
709 reflect the official policy or position of the U.S. government.

710 We thank Dr. Rusheng Zhang and Shengyin Shen of the University of Michigan for helping acquire vehicle  
711 trajectory data used in this paper.

## 712 **Author contributions**

713 X.Y. and H. L. conceived and led the research project. X. Y., Z. Z., and H. L. developed the framework and  
714 algorithm and wrote the paper. X. Y., S. F., and H. L. developed the statistical realism requirement for the  
715 simulation system. X. Y. and Z. Z. implemented the algorithms and prepared the experiment results. H. Z. and  
716 H. S. developed the SUMO experiment environment. H. Z. prepared the model scalability experiment results. All  
717 authors provide feedback during the manuscript revision and results discussions. H. L. approved the submission  
718 and accepted responsibility for the overall integrity of the paper.

## 719 **Competing interests**

720 The authors declare no competing interests.

## 721 **Figure legends/captions**

722 **Figure 1. Modeling naturalistic driving environment with statistical realism.** **a**, Statistical errors in simulation  
723 may mislead AV development. **b**, The underlying naturalistic driving environment distribution is highly complex  
724 and in a high dimensional space since it involves multiple agents and long time horizons. The simulation  
725 environment needs to achieve statistical realism, i.e., distribution-level accurate statistics regarding human driving  
726 behaviors in both normal and safety-critical driving conditions. **c**, Major challenges for modeling multi-agent  
727 interaction behaviors and constructing naturalistic driving environments. The challenges include the “curse of  
728 dimensionality” for multi-agent highly interactive behaviors, the “curse of rarity”<sup>40</sup> of safety-critical events in the  
729 real world, and the “distribution shift” for long-time simulations.

730 **Figure 2. The proposed NeuralNDE framework.** **a**, The framework and training pipeline of the NeuralNDE.  
731 Two types of loss, i.e., imitation loss and adversarial loss, will be backpropagated to train the behavior modeling  
732 network to learn multi-agent interactive behaviors. The adversarial loss will also be used to train the discriminator  
733 for distinguishing between real-world and simulated trajectories. The safety mapping network is pretrained and  
734 fixed when training the behavior modeling network and the discriminator jointly. **b**, Demonstration of the  
735 behavior modeling network, conflict critic module, and safety mapping network during the inference time. The  
736 behavior modeling network predicts the action distribution of each vehicle. In safety-critical situations,  
737 predictions can be inaccurate and lead to unrealistically high crash probability, because the training data is  
738 overwhelmingly from normal driving situations. Therefore, the safety mapping network will guide vehicle  
739 behavior and rectify their actions in such situations. With a certain probability, we will accept the generated  
740 conflict without passing it to the safety mapping network. The acceptance probability is trajectory-dependent and  
741 will be calibrated to fit ground-truth safety-critical statistics (e.g., crash rate and crash type distribution). Therefore,  
742 the conflict critic module controls the occurring frequencies and patterns of dangerous driving behavior during  
743 the simulation. **c**, Illustration of the simulation process. The historical states of all road users will pass through  
744 the behavior modeling network, conflict critic module, and safety mapping network to generate the states at the  
745 next time step. This process will proceed progressively to simulate the driving environment.

746 **Figure 3. Illustration figure of the studied location.** **a**, Illustration figure of the Ann Arbor roundabout. **b**,  
747 Illustration figure of the yielding area. The vehicle in the red yielding area is the yielding vehicle. The vehicle in

748 the corresponding conflicting quadrant is the conflicting vehicle. The yielding distance and yielding speed are  
749 demonstrated in the figure.

750 **Figure 4. Statistical realism of normal driving behavior.** **a**, Vehicle instantaneous speed distribution. **b**,  
751 Vehicle distance distribution. **c**, Yielding distance distribution: distance between the yielding vehicle and its  
752 nearest conflicting vehicle. **d**, Yielding speed distribution: speed of the nearest conflicting vehicle.  $D_H$  and  $D_{KL}$   
753 denote the Hellinger distance and the KL-divergence, respectively.

754 **Figure 5. Statistical realism of safety-critical driving behavior.** **a**, Vehicle crash type distribution. **b**, Vehicle  
755 crash severity distribution. **c**, Vehicle distance distribution in near-miss situations. **d**, Post-encroachment time  
756 (PET) distribution in near-miss situations.  $D_H$  and  $D_{KL}$  denote the Hellinger distance and the KL-divergence,  
757 respectively.

758 **Figure 6. Crash events in the real world and NeuralNDE.** Each subfigure denotes a different crash type. The  
759 main image in each subfigure demonstrates the crash event generated by NeuralNDE. The image in the red box  
760 denotes the real-world crash event captured by roadside cameras. **a**, Angle crash caused by failure to yield.  
761 Vehicle #2 (shown in pink) fails to yield to vehicle #1 (shown in blue) and causes the collision. **b**, Sideswipe  
762 crash caused by improper lane usage. One vehicle improperly intrudes into the lane of another vehicle and causes  
763 the collision. **c**, Rear-end crash caused by failure to stop within assured clear distance. Vehicle #2 (shown in pink)  
764 fails to keep a safe distance from vehicle #1 (shown in blue) and causes the collision.

765 **Figure 7. Proof-of-concept for modeling a road network.** It involves two scenarios, i.e., a four-way intersection  
766 and a two-lane roundabout, The areas denoted by red rectangles are controlled by NeuralNDE methods, and the  
767 areas denoted by yellow rectangles are controlled by model-based methods.

768 **Figure 8. Statistical realism of the intersection area in the road network.** **a**, Vehicle instantaneous speed  
769 distribution. **b**, Vehicle distance distribution. **c**, Vehicle distance distribution in near-miss situations. **d**, Post-  
770 encroachment time (PET) distribution in near-miss situations.  $D_H$  and  $D_{KL}$  denote the Hellinger distance and the  
771 KL-divergence, respectively.

772 **Figure 9. Statistical realism of the roundabout area in the road network.** **a**, Vehicle instantaneous speed  
773 distribution. **b**, Vehicle distance distribution. **c**, Yielding speed distribution. **d**, Yielding distance distribution. **e**,  
774 Vehicle distance distribution in near-miss situations. **f**, Post-encroachment time (PET) distribution in near-miss  
775 situations.  $D_H$  and  $D_{KL}$  denote the Hellinger distance and the KL-divergence, respectively.

776 **Figure 10. Illustration of individual modules.** **a**, Network architecture of the behavior modeling network. **b**,  
777 The illustration of the conflict critic module.