Research Article

Transportation Research Record
1–11

DOI: 10.1177/03611981211035756
journals.sagepub.com/home/trr

Learning-Based Stochastic Driving Model for Autonomous Vehicle Testing

Lin Liu[1,2], Shuo Feng[2], Yiheng Feng[3], Xichan Zhu[1], and Henry X. Liu[2,4]

Abstract

In the simulation-based testing and evaluation of autonomous vehicles (AVs), how background vehicles (BVs) drive directly influences the AV's driving behavior and further affects the test results. Most existing simulation platforms use either predetermined trajectories or deterministic driving models to model BV behaviors. However, predetermined BV trajectories cannot react to AV maneuvers, and deterministic models are different from real human drivers because of the lack of stochastic components and errors. Both methods lead to unrealistic traffic scenarios. This paper presents a learning-based stochastic driving model that meets the unique needs of AV testing (i.e., interactive and human-like stochasticity). The model is built based on the long short-term memory architecture. By incorporating the concept of quantile regression into the loss function of the model, the stochastic behaviors are reproduced without prior assumption of human drivers. The model is trained with the large-scale naturalistic driving data (NDD) from the Safety Pilot Model Deployment project and compared with a stochastic intelligent driving model (IDM). Analysis of individual trajectories shows that the proposed model can reproduce more similar trajectories of human drivers than IDM. To validate the ability of the proposed model in generating a naturalistic driving environment, traffic simulation experiments are implemented. The results show that traffic flow parameters such as speed, range, and time headway distribution match closely with the NDD, which is of significant importance for AV testing and evaluation.

Keyword

Data and Data Science, Artificial Intelligence and Advanced Computing Applications, Machine Learning (Artificial Intelligence), Statistical Methods, Safety, Operations, Vehicle-Highway Automation

Testing and evaluation of autonomous vehicles (AVs) has become an active research topic in the past few years (*1–7*). Among three major testing methods (simulation, test track, and on-road) (*8*), simulation is the most cost-effective, efficient, and safe method, which attracts significant attention especially in the early development stage of AVs (*9–12*). To evaluate the performance of AV models in a simulation environment, background vehicles (BVs) need to be generated to interact with the AV models in different testing scenarios (*13*). To model driver behaviors for the purpose of AV testing, the following features should be included:

1. Interactive: The BVs should react to the AV behavior in real time. When considering AV testing in a realistic driving environment such as highway driving environments, the interactive behaviors will affect the test results. For example, the interactivity is significant for functions like overtaking, highway merging, and unprotected left turn.
2. Human-like: The BVs should act like a human driver with stochastic components and errors. For example, different driving styles or mental

[1]School of Automotive Studies, Tongji University, Shanghai, China
[2]Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI
[3]Lyles School of Civil Engineering, Purdue University, West Lafayette, IN
[4]University of Michigan Transportation, Research Institute, Ann Arbor, MI

Corresponding Author:
Shuo Feng, fshuo@umich.edu

states of a driver may lead to variation in driving behavior even in the same traffic environment.

In the past few years, two main methods have been proposed to model BV behaviors for AV testing. In the first method, BV behaviors are defined before the testing. The method is firstly used for testing advanced driver assistant systems like adaptive cruise control and autonomous emergency braking. The speed profile of the leading vehicle is predefined with a combination of constant speed, acceleration, or jerk to generate the testing matrix (*14–16*). However, the produced trajectories are not human-like as human drivers usually behave much more stochastically. Another commonly used approach to apply predefined trajectories is to utilize the real-world data collected by vehicles equipped with multiple sensors to replicate the testing scenarios (*17*). Although the behaviors of human drivers can be precisely captured, the main problem of using predefined BV trajectories is that the testing is not interactive, because BVs cannot adjust their maneuvers dynamically based on AV behaviors.

The second method models BV behaviors with microscopic traffic flow models, which is an interactive approach. Based on certain driver behavior rules, the motion of each BV at each simulation time step can be updated according to the current traffic state. However, most existing driving behavior models are deterministic, such as Newell's (*18*) model, intelligent driving model (IDM) (*19*), and Gipps' (*20*) model, which cannot capture real human driver behaviors. Consequently, the AV may pass the test by just "remembering" the experienced scenarios.

Recently, several studies have focused on modeling stochastic driving behaviors by adding noise to existing deterministic models. Based on Newell's model, Treiber et al. (*21*) proposed a stochastic desired acceleration model and added Gaussian noise to the driver's desired acceleration. Laval et al. (*22*) extended the IDM model (*19*), optimal velocity model (*23*), and velocity difference model (*24*) by adding Gaussian noise to the driver's desired acceleration. Most existing models introduce a noise term to a deterministic model to achieve a stochastic driving behavior. The noise added is commonly the Gaussian noise (*21–24*). However, as the distribution of the driving behavior has been proven to be asymmetric (*25–27*), applying symmetric Gaussian distribution to describe the stochasticity of the driving behavior is questionable. To the best of the authors' knowledge, most of these models were designed for traffic flow analysis, and there is no existing model designed to capture the stochastic human driving behaviors, especially for AV testing.

There are two recent works designed to meet the unique requirements of driving models for AV testing. Yang and Peng (*26*) developed a car-following model with perceptual limitation, time delay, and distraction of human drivers. These effects were modeled as influencing parameters to design a stochastic model. However, the fixed types of effects limited the accuracy of the model and some human-related parameters are difficult to obtain. Feng et al. (*13*) and Yan et al. (*28*) proposed an empirical method to generate a naturalistic testing environment for AV testing. Stochasticity is introduced by sampling the driver behavior from the corresponding behavior distribution. However, as the empirical model is essentially a large-scale table, it suffers from discretization errors.

This paper proposes a learning-based stochastic driving model to meet the unique needs (i.e., interactive and human-like stochasticity) for the assessment of AVs. The goal of the model is to generate the action distribution that is consistent with the naturalistic driving data (NDD), given current vehicle states. Then by sampling the action from the distribution at each time step, the model can interact with the AV in a stochastic and realistic way. To achieve this goal, the quantile regression (QR) (*29*) method is incorporated into the learning process. Instead of one single output (e.g., expected acceleration) as commonly designed in existing methods, this study method provides a series of outputs, which are designed as the different quantiles of actions. Correspondingly, the pinball loss function (*29*) is applied to calculate loss. By decreasing the loss function, the output actions learn to be the quantiles, which can fit the NDD via the kernel density estimation (KDE) (*30*). To further capture the temporal dependency of the behavior model, the long short-term memory (LSTM) (*31*, *32*) recurrent neural network (RNN) architecture is utilized. The proposed method is referred to as QRLSTM hereafter in this paper. To validate the effectiveness of the proposed method, a large-scale naturalistic driving database is utilized from the Safety Pilot Model Deployment (SPMD) project (*33*). Simulation results show that the proposed QRLSTM model can represent human driving behaviors at both microscopic and macroscopic levels, which greatly enhances the previous studies by providing a human-like interactive driving environment for AV testing and evaluation.

The new model has two significant advantages. First, it does not apply any assumption of the distribution of human driving error or require any prior knowledge of human drivers. With the QR model structure and KDE, the stochasticity of human driving is obtained in a data-driven way. Second, the model has the ability to generate a realistic driving environment, which is of significant value for AV testing and evaluation.

The rest of this paper is organized as follows. The method section formulates the modeling problem and describes the structure of the QRLSTM model. The next section introduces the model training process. After that, simulation experiments are presented, followed by the results and discussion. Conclusions and further research are provided in the final section.

Method

Problem Formulation

In the introduction section, two features (i.e., interactive and human-like stochasticity) are proposed for BVs to generate a realistic traffic environment for AV testing. The behavior modeling problem of BVs is formulated as follows.

Interactive Driving Behavior. A microscopic driving model can realize the interaction of the BV with AV as well as other BVs. The goal of a microscopic driver behavior model f_d is to calculate or predict the action (e.g., speed or acceleration) of the BV $\hat{y}_{t+1}$ at time step $t+1$, given the traffic state x_t at time step t, that is,

$$\hat{y}_{t+1} = f_d(x_t), \tag{1}$$

where traffic state x_t refers to the dynamic state of BV and all vehicles around it, including AV. In an AV testing simulation, the action of the AV will change the traffic state and BV will calculate the next action accordingly. In this way, the microscopic driving model realizes the interaction between BVs and the AV.

In the car-following situation, the traffic state x typically only considers the velocity of the BV v, the velocity of the AV v^l, the bumper-to-bumper range between the BV and the AV r, the range rate between the BV and the AV rr, and acceleration of the BV a, that is, $x = [v, v^l, r, rr, a]$. The action could be either the velocity or acceleration of the BV, denoted as v or a. Therefore a car-following model could be written as

$$v_{t+1} = f_d(v_t, v_t^l, r_t, rr_t, a_t), \tag{2}$$

$$\text{or } a_{t+1} = f_d(v_t, v_t^l, r_t, rr_t, a_t). \tag{3}$$

For example, the IDM model has the following form:

$$a_{t+1} = a[1 - (\frac{v_t}{v0})^4 - (\frac{s0 + v_t T + \frac{v_t rr_t}{2\sqrt{ab}}}{r_t})^2], \tag{4}$$

where $v0, s0, a, b, T$ are constants (*19*). Therefore, the IDM model could be simplified as

$$a_{t+1} = f_{IDM}(v_t, r_t, rr_t). \tag{5}$$

Stochastic Driving Behavior. In actual driving scenarios, human drivers do not behave deterministically as described in the previous section, resulting in the need for a stochastic microscopic driving model. With this model, the action of the vehicle might be various even given the same traffic state as input. Several studies have tried to build such a model by adding simple noise (e.g., Gaussian noise) to existing deterministic models. For example, the output of the IDM model is added with a Gaussian noise term in Treiber et al. (*21*). The modified IDM model could be written as

$$a_{t+1} = f_{IDM}(v_t, r_t, rr_t) + \sqrt{Q}\xi_\alpha(t), \tag{6}$$

where Q is the fluctuation strength and $\xi_\alpha(t)$ is the Gaussian noise. The method of adding Gaussian noise to an analytic deterministic model has two disadvantages: the assumption of Gaussian distribution may not reflect real human stochastic driving behaviors, and the analytic form will disenable the model to fit the changing driving behavior.

To release the unrealistic assumption of the Gaussian distribution of randomness, a new model structure is proposed in this paper. Instead of calculating the action, the model directly outputs the distribution of action and the final action is sampled accordingly. The model is defined as

$$F(\hat{y}_{t+1}) = f_d(x_t) = f_d(v_t, v_t^l, r_t, rr_t, a_t), \tag{7}$$

where $F()$ denotes the distribution function. To achieve such a model structure, the LSTM is modified with QR loss and KDE. The main advantage of this model structure is that no prior assumption is applied on the distribution form. Instead, the distribution is directly learned and estimated from real driving data. The structure of the model is described in detail in the following sections.

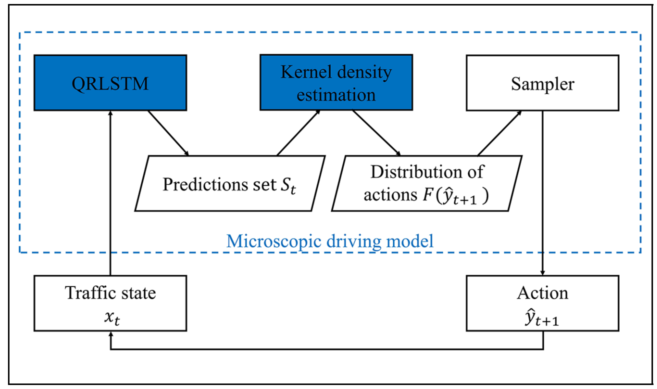

Figure 1. The overall model framework.
Note: QRLSTM = quantile regression long short-term memory.

Model Framework

Figure 1 shows the overall framework of the proposed QRLSTM model, which contains three components:

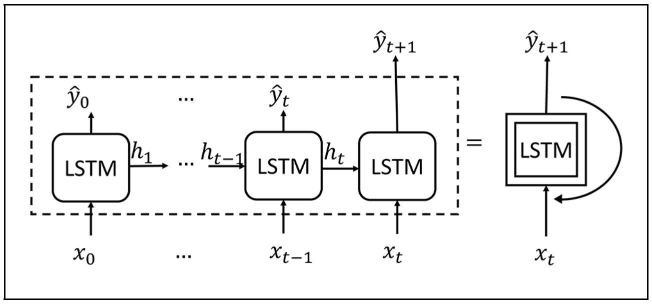

Figure 2. Illustration of the LSTM schema.
Note: LSTM = long short-term memory.

QRLSTM, KDE, and a sampler. Given the traffic state x_t, the QRLSTM model outputs a set of predicted actions S_t according to the quantile definition P. Then, these actions will be used in KDE to estimate the continuous action distribution F. Finally, action $\hat{y}_{t+1}$ is sampled from the F distribution. The integration of QRLSTM, KDE, and the sampling process forms a stochastic microscopic driving model. The action $\hat{y}_{t+1}$ is used to update the traffic state x_{t+1} at time $t+1$, and the loop runs repeatedly.

LSTM Structure. A significant trend in driver behavior modeling is utilizing machine learning techniques to take advantage of real-world driving data. Neural networks were introduced to model car-following in Hongfei et al. (*34*) and improved by adding human factors by Khodayari et al. (*35*). Different learning-based model structures further improve the modeling performance, such as the deep neural networks model built by Wang et al. (*32*) and the reinforcement learning model built by Zhu et al. (*36*). In Zhou et al. (*31*) and Wang et al. (*32*), the RNN model, which can take the historical state into consideration, shows better performance of the mean square error (MSE) in speed prediction.

In this paper, the LSTM neural network, a widely used neuron network structure, is applied as the base model to calculate the BV action, although the present framework is applicable for generic neuron network structures. A simple illustration of LSTM is shown in Figure 2. The detailed neuron structure can be found in Hochreiter and Schmidhuber (*37*). To calculate $\hat{y}_{t+1}$, LSTM considers both the current input x_t and the hidden state h_t, which is calculated based on x_{t-1} and h_{t-1}, respectively. With this structure, the LSTM can learn both the corresponding output and the hidden sequence patterns with sequential training data.

QRLSTM Structure. Owing to the randomness and error of human drivers, the action is various even in the same traffic state. Although the driver behavior is recorded as state-action pairs in real-world data, there is an action distribution under a certain traffic state. However, the LSTM is a deterministic model that outputs one action given a traffic state. As shown in Figure 3, given traffic state x_t, LSTM outputs the action $\hat{y}_{t+1}$. The error is then calculated for MSE for adjusting model weights. Training the model with the MSE cost function will indeed lead the model to estimate the median of actions in a traffic state, which will lose the stochastic information of real-world data.

To capture the stochasticity of driver behaviors, applying the concept of quantile regression (*29*) is proposed. As shown in Figure 3, the main differences between QRLSTM and LSTM are the output forms of the models and the loss functions used for model training. Specifically, a QRLSTM model outputs a set of N actions $\hat{y}_{t+1,1}, \hat{y}_{t+1,2} \ldots \hat{y}_{t+1,N}$. The error is calculated as the pinball error and used for adjusting the model weights. By applying the pinball loss as the loss function of LSTM, the target of training is changed to estimate action quantiles in a traffic state, instead of the action median.

The pinball function (*38*) is designed to calculate the error of a quantile to the real value. The pinball function is defined as

$$L_{p,y_t} = \begin{cases} p(y_t - \hat{y}_{t,p}) & if(y_t - \hat{y}_{t,p}) \geq 0 \\ (p-1)(y_t - \hat{y}_{t,p}) & if(y_t - \hat{y}_{t,p}) < 0 \end{cases}, \quad (8)$$

where $0 < p < 1$ is the quantile probability, y_t is the observed output from data, $\hat{y}_{t,p}$ is the prediction of $p-$ quantile, and L_{p,y_t} is the loss of the predicted $p-$ quantile for y_t. The loss of the QRLSTM model to estimate the $p-$ quantile is then defined as

$$L_p = \frac{1}{N} \sum_{t=2}^{N+1} L_{p,y_t}, \quad (9)$$

where N is the total number of y_t.

Then, QRLSTM is designed to calculate an action matrix corresponding to a set of quantile probabilities, $p \in P$. $P = \{\frac{1}{|p|-1}, \frac{2}{|P|-1}, \ldots, 1 - \frac{1}{|P|-1}\}$, where $|P|$ is the length of P, that is, the number of quantile probabilities p. The loss function of the QRLSTM model is set as

$$L = \frac{1}{N|P|} \sum_{t=2}^{N+1} \sum_{\forall p \in P} L_{p,y_t}. \quad (10)$$

By training with real-world data, the action set will converge to the action quantiles.

KDE and Sampler. Given a traffic state x_t, QRLSTM predicts a $|P| \times 1$ quantile set $S_t = \{y_{t,p_1}, y_{t,p_2}, \ldots, y_{t,p_{|P|}}\}$. To obtain a continuous prediction distribution, KDE (*30*) is applied. As a classic non-parametric estimation method,

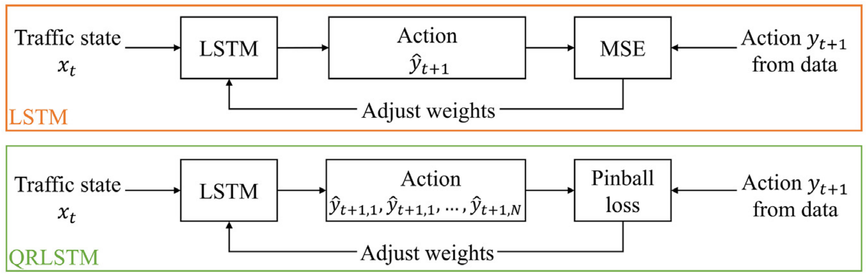

Figure 3. Differences between LSTM and QRLSTM.
Note: LSTM = long short-term memory; QRLSTM = quantile regression LSTM.

KDE does not require a prior assumption of distribution form, which is suitable for modeling the driver behaviors. The KDE estimation of S_t is calculated by

$$f(\hat{y}_t) = \frac{1}{B|P|} \sum_{\forall p \in P} K(\frac{\hat{y}_{t,p} - y_t}{B}), \quad (11)$$

where $B>0$ is the bandwidth and K is a kernel function.

The sampler will then generate one random variate from F as the final output of the QRLSTM model. The sampling method here is the rejection sampling method (*39, 40*) with two steps: first, obtain a sample y from the reference Gaussian distribution with density g and a sample u from the uniform distribution over the unit interval. Second, accept y as a sample if $u < \frac{f(y)}{g(y)}$. Otherwise reject the value of y and return to the first step.

Model Training

This paper focuses on one of the most common driver behaviors, car-following behavior, to demonstrate the proposed model. To capture human driving patterns, the QRLSTM model is trained with real driving data from a naturalistic driving study. In real traffic environments, different types of drivers have diverse driving behavior styles, therefore, different models are trained for different drivers to capture each driver's behaviors.

Data Description

NDD from the SPMD project (*33*) was adopted to train and test the proposed model. With 2,842 participating vehicles, the SPMD project collected NDD of over 34.9 million miles in Ann Arbor, Michigan. The data from 86 vehicles equipped with data acquisition systems were used, including MobileEye cameras that capture vehicle trajectory data (e.g., position, speed, etc.) of the vehicle and its surrounding traffic (e.g., leading vehicles in the same and adjacent lanes) with a frequency of 10 Hz. To obtain data in car-following scenarios, the following filtering criteria are applied:

- Road type is highway.
- Speed is larger than 20 m/s (72 km/h)
- A leading vehicle is identified by the data acquisition system.

Finally, a total number of 24,816 trajectories are extracted with a total travel time of 1,403,955 s (around 390 h). Each trajectory lasts for 1 to 1,768 s randomly.

Model Settings

In the car-following scenario, the traffic state x includes the velocity of the BV v, the velocity of the leading vehicle v^l, range between the BV and the AV r, and range rate rr. The action is the acceleration of the BV a. The memory time of the LSTM is set as 10, that is, the input of the model is the traffic states from the previous 1 s, namely, $x_t = \{v_{t-9}, v_{t-8}...v_t; v^l_{t-9}, v^l_{t-8}...v^l_t; r_{t-9}, r_{t-8}...r_t; rr_{t-9}, rr_{t-8}...rr_t\}$. The QRLSTM model has three hidden layers with 32 LSTM neurons. The quantile probability P is set as $P = \{0.05, 0.10...0.95\}$. The bandwidth B in KDE is 0.75, and K is the Gaussian kernel. This model setting is implemented on all QRLSTM models in this paper.

Model Training

Although there are data from 86 drivers available from SPMD, not all of them have enough data to train a QRLSTM model. The total driving time of each driver is shown in Figure 4. There is a large variance in travel time among different drivers. To confirm the required data for the training process, the QRLSTM model is trained with 0.01%, 0.05%, 0.1%, 0.5%, 1%, 10%, 25%, 50%, and 70% of all available data, respectively, to help investigate how the validation error of the model changes along with the data size. To calculate the pinball loss as the validation error, 5% of all available data are randomly sampled

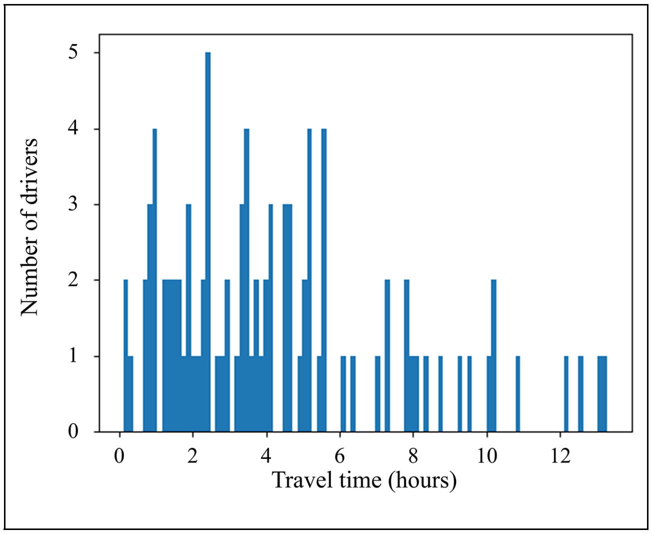

Figure 4. Distribution of total travel time of 86 drivers.

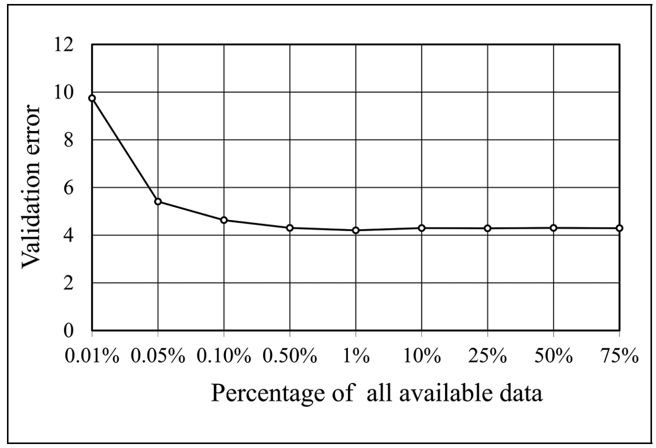

Figure 5. Validation error versus size of training data.

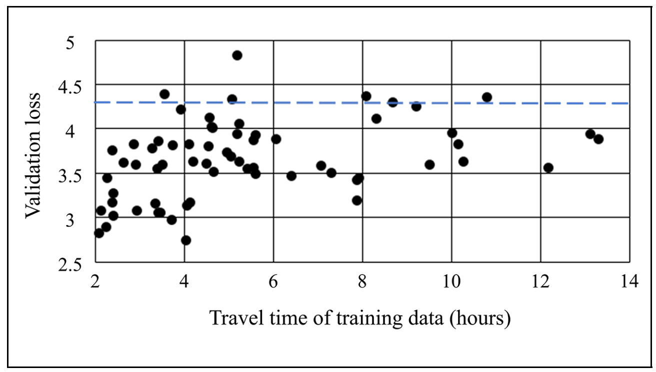

Figure 6. Validation loss.

as validation data. As shown in Figure 5, the validation error drops quickly as the size of the training data grows. The decreasing trend slows down when 0.5% of total data are used in the training, which accounts for about 7,019 s (about 2 h) driving time. Therefore, 2 h is set as the threshold of necessary data size, and 66 drivers satisfying the condition are modeled individually.

The training results of the obtained 66 QRLSTM models are shown in Figure 6. The average validation loss of individual models is 3.65. The blue dash line stands for the validation loss of the mixed model trained with data of all drivers. Compared with the validation loss of the mixed model, only six individual driver models converge to a higher loss; that is, 90.91% models outperform the mixed model. The underperformance of the mixed driver against individual models is owed to various driving styles and habits of different drivers.

Simulation Experiments

Experiments Setup

To verify the performance of the proposed method, simulation experiments are designed and conducted with Python 3.7, in a workstation equipped with Intel i7-10700K CPU and 16 GB RAM. As only car-following behavior is considered, a 3-mile single-lane highway is built. The traffic demand varies from 500 to 2,000 vehicles/hour in different experiments. The experiment lasts for 1 h and the simulation resolution is 10 Hz. The workflow of the simulation is shown in Figure 7.

When the simulation starts, the traffic initializer generates the first two vehicles in the network, by randomly sampling the speed of the leading vehicle v_0^l, range r_0, and the speed of the following vehicle v_0 from the SPMD, and then calculating the corresponding range rate rr_0. It forms the traffic state at time step 0, that is, $x_0 = v_0^l, v_0, r_0, rr_0$. As the LSTM model structure needs the traffic state of the last 10 time steps as the input, the IDM is used to generate the traffic state of the following 9 time steps, to obtain the traffic state of the first 10 time steps.

After the initialization, given traffic state at time t for each vehicle model, $s_t = x_{t-9}, x_{t-8}...x_t$, the vehicle model predicts its action a_t. Then with action a_t, the state updater calculates the traffic state for time $t+1$ in the sequence of vehicle positions. The vehicle generator decides whether a new vehicle should be generated and join the traffic flow. The decision is made by binomial sampling with the probability calculated according to current traffic demand. If a new vehicle is generated, a trained driver model from the 86 drivers will be randomly assigned to the new vehicle according to the travel time distribution as shown in Figure 4.

Results and Discussion

As discussed in the introduction, to act as BVs in AV testing scenarios, the driver model should interact with

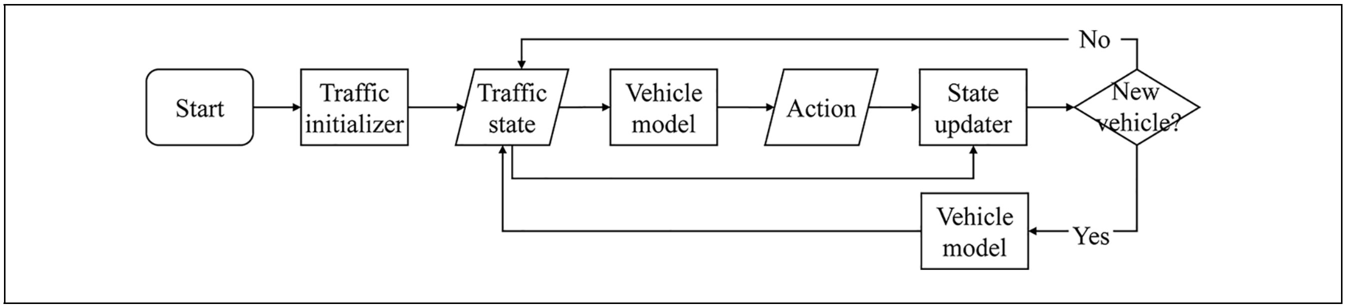

Figure 7. Workflow of mixed driver simulation experiment.

Table 1. Parameters of the Modified IDM

Parameter		Value
v_0	Desired velocities	34.99 m/s
s_0	Minimum gap	1.70 m
a	Acceleration	1.5 m/s^2
b	Comfortable deceleration	0.66 m/s^2
T	Desired time headway	0.73 s
Q	Fluctuation strength	0.001 m^2/s^3

Note: IDM = intelligent driving model.

the AV, which is already satisfied with the QRLSTM structure and the simulation workflow. Another requirement is that the model should be consistent with stochastic human driving behaviors to construct a realistic driving environment. Therefore, performances on both trajectory level and traffic level need to be examined to see whether the vehicle trajectories generated from the proposed model match the human drivers in the NDD. For comparison, the model proposed by Treiber et al. (*21*) is replicated as the baseline, which is an extension of the commonly used IDM by adding Gaussian noise to the driver's acceleration. The model is defined in Equations 3 and 4 and the model parameters are also calibrated with the SPMD dataset for a fair comparison, shown in Table 1.

Trajectory Reproducing Accuracy. Figure 8 shows a comparison at the individual trajectory level among the ground truth, QRLSTM, and the modified IDM model. The maximum simulation time is set as 30 s and both models run a total of 100 trajectories. The figure represents four examples of the generated continuous action profiles given the initial condition (i.e., traffic states for the first second). The red, green, and blue lines stand for the real vehicle action profile obtained from the NDD, the proposed QRLSTM model, and the modified IDM model, respectively. In each subfigure, the acceleration and speed profiles of the same trajectory are represented. The figure shows a similar ability of two models to capture the behavior trend of human drivers. To quantitatively compare the trajectory-reproducing ability of two models, the MSE of acceleration and speed are calculated. The MSEs of the acceleration of QRLSTM and IDM are 0.0501 and 0.0619, respectively. The MSEs of the speed of QRLSTM and IDM are 3.4992 and 4.2518, respectively. The statistical comparison indicates that the proposed QRLSTM outperforms IDM in both acceleration and velocity.

Traffic Parameters Comparison. To compare the driving environment generated by the proposed model and the real driving environment, the distributions of speed, range, and time headway (THW) are selected as measurements to describe the traffic flow. Simulations described in the section of simulation experiments are conducted with the proposed QRLSTM and IDM models. A total of 1,159 km of vehicle trajectories are generated by simulation with QRLSTM, a total of 1,789 km of vehicle trajectories are generated by modified IDM, and 100,000 data points are randomly selected from SPMD to form the ground truth of distributions. As shown in Figure 9, the red, green, and blue distributions represent real driving data from SPMD, trajectories from the proposed QRLSTM model, and the modified IDM, respectively.

Speed and range are two major parameters that can describe car-following behaviors, which are critical for AV testing. The upper two subfigures in Figure 9 present the speed and range distributions, respectively. Both distributions of QRLSTM are much more similar to the SPMD than to IDM in the form of distribution and the median of the parameter. For speed distributions, IDM is much more concentrated than QRLSTM and the ground truth, indicating that the vehicles controlled by IDM have a similar driving pattern with each other. In the meantime, the speed distribution of QRLSTM demonstrates stochastic driving behaviors. For range distributions, the median range of IDM is much greater than that of QRLSTM and the ground truth, indicating a more conservative driving style. Furthermore, distributions of THW are also compared, which is critical for the

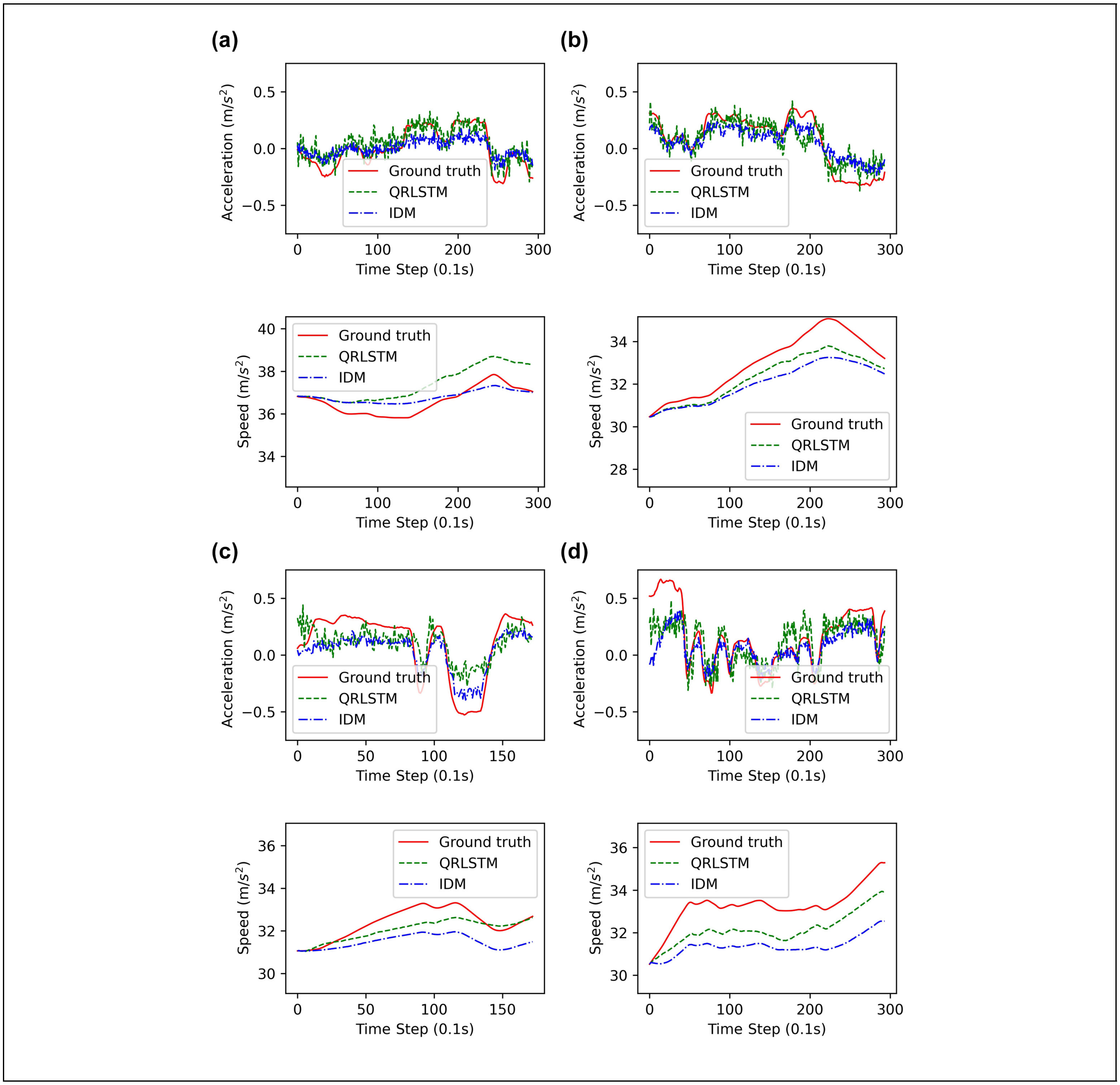

Figure 8. Four trajectory reproducing examples.
Note: QRLSTM = quantile regression long short-term memory; IDM = intelligent driving model.

safety test of AVs. As shown in the lower figure in Figure 9, the THW distribution of the QRLSTM model can also better capture the trend of the real-world driving environment than IDM.

To compare the performance, cross-entropy is introduced as a numerical measurement of the similarity between two distributions. The cross-entropy of distribution q relative to distribution p within the same sample space X is defined as follows:

$$H(p,q) = -\sum_{x \in X} p(x) log q(x). \tag{12}$$

The smaller the cross-entropy is, the more similar the two distributions are. As shown in Table 2, the cross-entropies of all three distributions of the QRLSTM model and NDD are significantly smaller than those of the modified IDM and NDD. It means that the distributions of speed, range, and THW of the QRLSTM model

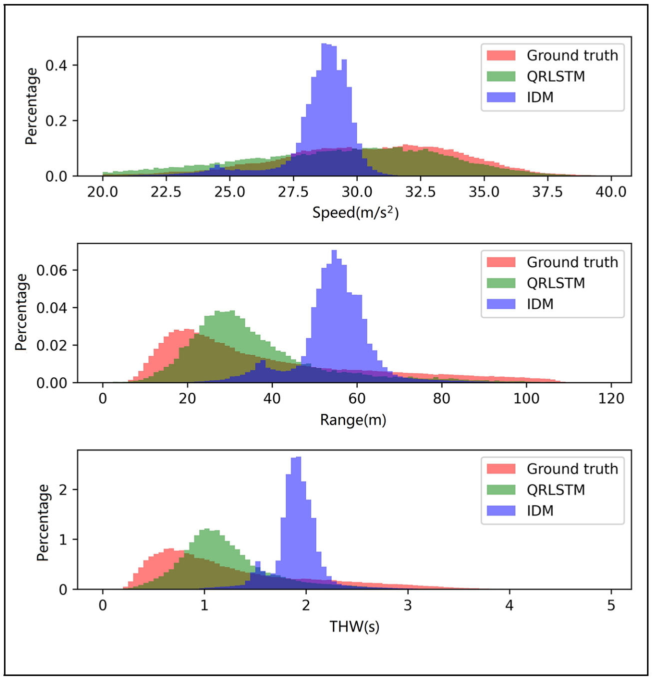

Figure 9. Traffic parameter comparisons between QRLSTM, IDM, and NDD.
Note: QRLSTM = quantile regression long short-term memory; IDM = intelligent driving model; NDD = naturalistic driving data; THW = time headway.

Table 2. Cross Entropy of QRLSTM and IDM with NDD

Model	Speed	Range	THW
QRLSTM and NDD	5.86	1.65	11.23
IDM and NDD	10.99	2.38	33.27
NDD and NDD	5.76	1.57	8.26

Note: QRLSTM = quantile regression long short-term memory; IDM = intelligent driving model; NDD = naturalistic driving data; THW = time headway.

are more similar to real-world distributions. Therefore, the proposed QRLSTM model outperforms IDM in generating a more realistic traffic environment.

Conclusion and Further Research

This paper has proposed a learning-based stochastic driving model structure for generating a realistic driving environment for AV testing and evaluation. Starting from the well-studied LSTM model structure, the model introduces stochasticity from the QR-based loss function without any assumption on the distribution of human driver behavior. The model is trained with real driving data from SPMD and compared with a modified IDM model, showing its superiority over traditional car-following models such as IDM. A microscopic comparison between individual trajectories shows that the proposed model is able to capture frequent variations of human driving. Moreover, a comparison at the macroscopic level shows that the speed, range, and THW distributions of the proposed model match the NDD distributions well. The results indicate that the traffic environment generated by the proposed model can reflect human driving behaviors.

This study has several limitations. First, this model is only applied to the car-following scenario in this paper and needs to be extended to scenarios that include lateral vehicle maneuvers such as lane changing and cut-in, where the interactions between AVs and BVs are more complex. Second, as the crash and near-crash scenarios are of great value for AV testing, whether the generated traffic environment can conduct these critical scenarios is another important feature in the naturalistic driving environment model. Third, how to integrate physical knowledge into the learning methods deserves more investigation. Recently, the concept of physics regularized machine learning has been proposed for macroscopic traffic flow modeling (*41*, *42*), which is also a promising direction for microscopic behavior modeling. By introducing physical knowledge of analytical models like IDM and Gipps' model, the learning-based model might facilitate a fast training process, accuracy modeling, and convenient implementation.

Acknowledgments

The authors thank the U.S. Department of Transportation (U.S. DOT) Region 5 University Transportation Center: Center for Connected and Automated Transportation (CCAT) of the University of Michigan for funding the research.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: Lin Liu, Henry X. Liu, Yiheng Feng, Shuo Feng, Xichan Zhu; data collection: Yiheng Feng, Shuo Feng; analysis and interpretation of results: Lin Liu; draft manuscript preparation: Lin Liu, Shuo Feng, Yiheng Feng, Henry X. Liu. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The U.S. Department of Transportation (U.S. DOT) Region 5 University Transportation Center: Center for

Connected and Automated Transportation (CCAT) of the University of Michigan funded the research.

ORCID iDs

Shuo Feng https://orcid.org/0000-0002-2117-4427
Yiheng Feng https://orcid.org/0000-0001-5656-3222
Henry X. Liu https://orcid.org/0000-0002-3685-9920

References

1. Kalra, N., and S. M. Paddock. Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability? *Transportation Research Part A: Policy and Practice*, Vol. 94, 2016, pp. 182–193.
2. Li, L., X. Wang, K. Wang, Y. Lin, J. Xin, L. Chen, L. Xu, B. Tian, Y. Ai, J. Wang, and D. Cao. Parallel Testing of Vehicle Intelligence Via Virtual-Real Interaction. *Science Robotics*, Vol. 4, No. 28, 2019, Article eaaw4106.
3. Li, L., N. Zheng, and F.-Y. Wang. A Theoretical Foundation of Intelligence Testing and Its Application for Intelligent Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2020, pp. 1–10.
4. Koren, M., S. Alsaif, R. Lee, and M. J. Kochenderfer. Adaptive Stress Testing for Autonomous Vehicles. *Proc., 2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1–7.
5. Koren, M., A. Corso, and M. J. Kochenderfer. The Adaptive Stress Testing Formulation. *arXiv Preprint arXiv:2004.04293*, 2020.
6. Corso, A., P. Du, K. Driggs-Campbell, and M. J. Kochenderfer. Adaptive Stress Testing with Reward Augmentation for Autonomous Vehicle Validation. *Proc., 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 163–168.
7. Weng, B., L. Capito, U. Ozguner, and K. Redmill. Towards Guaranteed Safety Assurance of Automated Driving Systems with Scenario Sampling: An Invariant Set Perspective (Extended Version). *arXiv Preprint arXiv:2104.09595*, 2021.
8. Thorn, E., S. C. Kimmel, M. Chaka, B. A. Hamilton, et al. *A Framework for Automated Driving System Testable Cases and Scenarios*. Technical Report. United States Department of Transportation, National Highway Traffic Safety Administration, Washington, D.C., 2018.
9. Feng, S., Y. Feng, C. Yu, Y. Zhang, and H. X. Liu. Testing Scenario Library Generation for Connected and Automated Vehicles, Part I: Methodology. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 3, 2020, pp. 1573–1582.
10. Feng, S., Y. Feng, H. Sun, S. Bao, Y. Zhang, and H. X. Liu. Testing Scenario Library Generation for Connected and Automated Vehicles, Part II: Case Studies. *IEEE Transactions on Intelligent Transportation Systems*, 2020, pp. 1–13.
11. Feng, S., Y. Feng, H. Sun, Y. Zhang, and H. X. Liu. Testing Scenario Library Generation for Connected and Automated Vehicles: An Adaptive Framework. *IEEE Transactions on Intelligent Transportation Systems*, 2020, pp. 1–10. https://doi.org/10.1109/TITS.2020.3023668.
12. Feng, S., Y. Feng, X. Yan, S. Shen, S. Xu, and H. X. Liu. Safety Assessment of Highly Automated Driving Systems in Test Tracks: A New Framework. *Accident Analysis & Prevention*, Vol. 144, 2020, Article 105664.
13. Feng, S., X. Yan, H. Sun, Y. Feng, and H. X. Liu. Intelligent Driving Intelligence Test for Autonomous Vehicles with Naturalistic and Adversarial Environment. *Nature Communications*, Vol. 12, 2021, p. 748.
14. Moon, S., I. Moon, and K. Yi. Design, Tuning, and Evaluation of a Full-Range Adaptive Cruise Control System with Collision Avoidance. *Control Engineering Practice*, Vol. 17, No. 4, 2009, pp. 442–455.
15. Euro, N. *European New Car Assessment Programme (Euro NCAP)—Test Protocol—AEB VRU Systems Version 3.0.4*. Technical Report. 2021.
16. Arcidiacono, A. *ADAS Virtual Validation: ACC and AEB Case Study with IPG CarMaker*. PhD thesis. Politecnico di Torino, 2018.
17. Li, W., C. Pan, R. Zhang, J. Ren, Y. Ma, J. Fang, F. Yan, Q. Geng, X. Huang, H. Gong, et al. AADS: Augmented Autonomous Driving Simulation Using Data-Driven Algorithms. *Science Robotics*, Vol. 4, No. 28, 2019, Article eaaw0863.
18. Newell, G. F. A Simplified Car-Following Theory: A Lower Order Model. *Transportation Research Part B: Methodological*, Vol. 36, No. 3, 2002, pp. 195–205.
19. Treiber, M., A. Hennecke, and D. Helbing. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E*, Vol. 62, No. 2, 2000, p. 1805.
20. Gipps, P. G. A Behavioural Car-Following Model for Computer Simulation. *Transportation Research Part B: Methodological*, Vol. 15, No. 2, 1981, pp. 105–111.
21. Treiber, M., A. Kesting, and D. Helbing. Understanding Widely Scattered Traffic Flows, the Capacity Drop, and Platoons as Effects of Variance-Driven Time Gaps. *Physical Review E*, Vol. 74, No. 1, 2006, Article 016123.
22. Laval, J. A., C. S. Toth, and Y. Zhou. A Parsimonious Model for the Formation of Oscillations in Car-Following Models. *Transportation Research Part B: Methodological*, Vol. 70, 2014, pp. 228–238.
23. Bando, M., K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama. Dynamical Model of Traffic Congestion and Numerical Simulation. *Physical Review E*, Vol. 51, No. 2, 1995, p. 1035.
24. Jiang, R., Q. Wu, and Z. Zhu. Full Velocity Difference Model for a Car-Following Theory. *Physical Review E*, Vol. 64, No. 1, 2001, Article 017101.
25. Newell, G. F. Instability in Dense Highway Traffic: A Review. *Proc., 2nd International Symposium on the Theory of Traffic Flow*, London, 1963.
26. Yang, H.-H. and H. Peng. Development of An Errorable Car-Following Driver Model. *Vehicle System Dynamics*, Vol. 48, No. 6, 2010, pp. 751–773.
27. Li, L. and X. M. Chen. Vehicle Headway Modeling and Its Inferences in Macroscopic/Microscopic Traffic Flow Theory: A Survey. *Transportation Research Part C: Emerging Technologies*, Vol. 76, 2017, pp. 170–188.

28. Yan, X., S. Feng, H. Sun, and H. X. Liu. Distributionally Consistent Simulation of Naturalistic Driving Environment for Autonomous Vehicle Testing. *arXiv Preprint arXiv:2101.02828*, 2021.
29. Koenker, R., and K. F. Hallock. Quantile Regression. *Journal of Economic Perspectives*, Vol. 15, No. 4, 2001, pp. 143–156.
30. Terrell, G. R., and D. W. Scott. Variable Kernel Density Estimation. *The Annals of Statistics*, Vol. 20, No. 3, 1992, pp. 1236–1265.
31. Zhou, M., X. Qu, and X. Li. A Recurrent Neural Network Based Microscopic Car Following Model to Predict Traffic Oscillation. *Transportation Research Part C: Emerging Technologies*, Vol. 84, 2017, pp. 245–264.
32. Wang, X., R. Jiang, L. Li, Y.-L. Lin, and F.-Y. Wang. Long Memory Is Important: A Test Study on Deep-Learning Based Car-Following Model. *Physica A: Statistical Mechanics and its Applications*, Vol. 514, 2019, pp. 786–795.
33. Bezzina, D., and J. Sayer. *Safety Pilot Model Deployment: Test Conductor Team Report*. Report No. DOT HS 812 171. National Highway Traffic Safety Administration, Washington, D.C., 2014.
34. Hongfei, J., J. Zhicai, and N. Anning. Develop a Car-Following Model Using Data Collected By "Five-Wheel System." *Proc., 2003 IEEE International Conference on Intelligent Transportation Systems*, Shanghai, Vol. 1. IEEE, Shanghai, China, 2003, pp. 346–351.
35. Khodayari, A., A. Ghaffari, R. Kazemi, and R. Braunstingl. A Modified Car-Following Model Based on a Neural Network Model of the Human Driver Effects. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, Vol. 42, No. 6, 2012, pp. 1440–1449.
36. Zhu, M., X. Wang, and Y. Wang. Human-Like Autonomous Car-Following Model with Deep Reinforcement Learning. *Transportation Research Part C: Emerging Technologies*, Vol. 97, 2018, pp. 348–368.
37. Hochreiter, S. and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, Vol. 9, No. 8, 1997, pp. 1735–1780.
38. Koenker, R., and G. Bassett Jr. Regression Quantiles. *Econometrica: Journal of the Econometric Society*, Vol. 46, No. 1, 1978, pp. 33–50.
39. Radford, N. Slice Sampling. *The Annals of Statistics*, Vol. 31, 2003, pp. 705–767.
40. Wells, M. T., G. Casella, and C. P. Robert. Generalized Accept-Reject Sampling Schemes. In *A festschrift for herman rubin* (A. DasGupta, ed.), Institute of Mathematical Statistics, 2004, pp. 342–347.
41. Yuan, Y., X. T. Yang, Z. Zhang, and S. Zhe. Macroscopic Traffic Flow Modeling with Physics Regularized Gaussian Process: A New Insight into Machine Learning Applications. *arXiv Preprint arXiv:2002.02374*, 2020.
42. Shi, R., Z. Mo, K. Huang, X. Di, and Q. Du. Physics-Informed Deep Learning for Traffic State Estimation. *arXiv preprint arXiv:2101.06580*, 2021.

The views presented in this paper are those of the authors alone.