



A sequence to sequence learning based car-following model for multi-step predictions considering reaction delay

Lijing Ma, Shiru Qu*

School of Automation, Northwestern Polytechnical University, Xi'an 710072, China



ARTICLE INFO

Keywords:

Sequence to sequence (seq2seq)
Deep learning
Car-following
Reaction delay
Traffic flow

ABSTRACT

Car-following behavior modeling is of great importance for traffic simulation and analysis. Considering the multi-steps decision-making process in human driving, we propose a sequence to sequence (seq2seq) learning based car-following model incorporating not only memory effect but also reaction delay. Since the seq2seq architecture has the advantage of handling variable lengths of input and output sequences, in this paper, it is applied to car-following behavior modeling to memorize historical information and make multi-step predictions. We further compare the seq2seq model with a classical car-following model (IDM) and a deep learning car-following model (LSTM). The evaluation results indicate that the proposed model outperforms others for reproducing trajectory and capturing heterogeneous driving behaviors. Moreover, the platoon simulation demonstrates that the proposed model can well reproduce different levels of hysteresis phenomenon. The proposed model is further extended with spatial anticipation, which improves platoon simulation accuracy and traffic flow stability.

1. Introduction

Car-following behavior modeling is of great importance for microscopic traffic simulation and human driving behavior analysis. Car-following models have been studied with diverse approaches over decades. As introduced by the review articles (Brackstone and McDonald, 1999; Saifuzzaman and Zheng, 2014), numerous mathematical car-following models have been developed. However, these car-following models rely on mathematical formulas, and the elaborate calibration of formulas' parameters is needed before traffic simulation which is a vital but challenging process (Li et al., 2016). More recently, with the development of machine learning, car-following behavior can be extracted directly from mass field data. As a result, this kind of data-driven car-following model has gained increasing interests. Furthermore, as one of the prevailing machine learning methods, deep learning algorithm has been intensively applied in car-following behavior modeling (Morton et al., 2017; Zhou et al., 2017; Huang et al., 2018; Wang et al., 2018; Wang et al., 2019).

Since human driving characteristics, such as memory effect and reaction delay, can influence traffic dynamics, researchers incorporated them into car-following models to have human driving behavior better reproduced. It is recognized that the memory effect is indispensable and can be embedded in deep learning car-following models by defining the historical time steps of input variables (Huang et al., 2018; Wang et al., 2018; Wang et al., 2019). Reaction delay has been estimated and widely considered in mathematical (Saifuzzaman and Zheng, 2014) and data-driven car-following models (Jia et al., 2003; Khodayari et al., 2012;

* Corresponding author.

E-mail address: qushiru@nwpu.edu.cn (S. Qu).

(Papathanasopoulou and Antoniou, 2015; Zheng et al., 2013). However, reaction delay is not fully considered in recently developed deep learning car-following models, particularly is not distinguished with the memory effect, which needs deeper investigating. Furthermore, in human driving behavior, the driving decisions (acceleration, deceleration, and cruising) are made for several time steps simultaneously rather than step by step, which is similar to the machine translation that translates one sentence completely into another instead of word by word. Thus, this multi-steps decision-making property within the car-following behavior needs to be considered.

The heterogeneous driving behaviors mean that drivers tend to have different driving strategies under the same circumstances, including aggressive maneuver, timid maneuver, and normal maneuver. With the study of heterogeneity, Laval and Leclercq (2010), Laval (2011), Chen et al. (2012a), Chen et al. (2012b), Chen et al. (2014) investigate the hysteresis phenomenon in traffic oscillation and find that heterogeneity is closely related to traffic hysteresis phenomenon. Although traffic oscillation and hysteresis are reproduced with deep learning car-following models (Zhou et al., 2017; Huang et al., 2018; Wang et al., 2019), the analysis of heterogeneity captured by deep learning models might be insufficient.

This paper aims to bridge the gaps with the proposition of a sequence to sequence (seq2seq) learning based car-following model. We use the seq2seq architecture proposed by Sutskever et al. (2014) due to its advantage in memorizing historical information and making multi-step predictions. With this strength, the seq2seq car-following model considers not only the memory effect that has been incorporated into other deep learning car-following models but also the reaction delay to fully reflect the driving state in the current time step and the decision-making property. To verify the seq2seq model, we compare the trajectory simulation results of the car-following models. Furthermore, the ability of seq2seq models for capturing heterogeneous driving behaviors is analyzed by models comparison and platoon simulation.

The rest of this paper is organized as follows. Section 2 reviews the literature on car-following models and reaction delay and prepares the experimental data. Section 3 proposes the framework of seq2seq car-following model. Section 4 presents experiments and discussion, mainly including the seq2seq model training process, the comparison of models, the analysis of heterogeneous driving behaviors, and the platoon simulation reproducing traffic oscillation and hysteresis. Section 5 concludes our findings.

2. Background

2.1. Car-following models

2.1.1. Mathematical car-following models

Car-following models have been extensively studied over decades, which describe the longitudinal interactions of adjacent vehicles. The first car-following model was proposed by Pipes (Pipes, 1953) as early as almost seventy years ago. Since then numerous mathematical car-following models have been developed, which in terms of different behavioral perspectives can be classified into four categories, including stimulus-response models, desired measures models, psycho-physical models, and Newell's simplified models.

The stimulus-response framework was first developed by General Motors research laboratories. The most well known Gazis-Herman-Rothery (GHR) model (Chandler et al., 1958; Gazis et al., 1961) assumes that the subject vehicle responds to given stimuli which include relative speed and spacing from the leading vehicle. Based on GHR models, the stimulus-response models have been extensively developed (Herman, 1959; Lee, 1966; Ahmed, 1999; Koutsopoulos and Farah, 2012).

The desired measures models are based on the assumption that the driver of the following vehicle attempts to minimize the difference between the actual and desired situation, such as space headway, time gaps, and speed. Helly (1959) first introduced this idea and proposed the function of desired space headway. The safety distance model developed by Gipps (1981) which in terms of desired time gaps is widely used in simulation software. The optimal velocity model (OVM) proposed by Bando et al. (1995) assumes that an optimal safe velocity namely the desired speed determines the acceleration of the following vehicle according to the difference between this and the actual one. And, it has been developed to extended models (Helbing and Tilch, 1998; Lenz et al., 1999; Jiang et al., 2001; Davis, 2003; Gong et al., 2008; Peng and Sun, 2010) to be more consistent with reality. In particular, the Intelligent Driver Model (IDM) proposed by Treiber et al. (2000) is one of the most widely applied desired-measures based models, which takes both the desired space headway and the desired speed into consideration.

The psycho-physical models are also known as action point models which suggest the perceptual threshold to define the stimuli that a driver can perceive and react to. Wiedemann (1974) expressed the threshold as a function based on the relative speed and space difference. The improved Wiedemann model (Fellendorf and Vortisch, 2010) is applied to simulation software.

Newell's simplified models (Newell, 2002; Laval and Leclercq, 2010) assume that the trajectory of the following vehicle can be parsimoniously replicated from the leading vehicle by shifting in time and space. The models have been extensively developed to explain driver behaviors and capture traffic oscillations (Zheng et al., 2011; Chen et al., 2012a; Laval et al., 2014).

2.1.2. Data-driven car-following models

With the development of artificial intelligence and high-fidelity traffic data, machine learning approaches gain increasing popularity of its implementation for car-following behavior modeling. Neither mathematical formula nor calibration is needed to be concerned in data-driven car-following models, and car-following behavior can be extracted from mass field data. Nonparametric regression, support vector regression, and neural networks are three main data-driven approaches, and they are all reported to be outperforming mathematical models. Among them, the neural networks approach is prevailing in recent studies.

The main idea of nonparametric regression car-following models is that the traffic dynamics can be straightforwardly reproduced by

the prediction of vehicle positions. Locally weighted regression was first introduced by [Toledo et al. \(2007\)](#) to the estimation of vehicle trajectories and improved by [Papathanasopoulou and Antoniou \(2015\)](#). [He et al. \(2015\)](#) proposed a nonparametric car-following model with k-nearest neighbor and validated it by field data.

Support vector machine (SVM) is a universal learning method based on the statistical learning theory, and support vector regression (SVR) is an application of SVM for regression problems. [Wei and Liu \(2013\)](#) first introduced SVR to car-following modeling to investigate the asymmetric characteristic in car-following behavior, because of its superiority of discovering inherent relations among the variables in a dataset. This SVR car-following model takes space headway, follower's speed, and relative speed as inputs, and outputs the follower's speed.

The first neural networks based car-following model goes back to [Kehtarnavaz et al. \(1998\)](#), then researchers developed the car-following models based on artificial neural networks (ANNs) ([Jia et al., 2003; Panwai and Dia, 2007; Chong et al., 2011; Khodayari et al., 2012; Zheng et al., 2013](#)), which all belong to the conventional neural networks approach. On the other hand, deep learning algorithms are applied to car-following behaviors modeling and demonstrated to be superior to conventional neural networks. The recurrent neural networks (RNNs) based car-following model proposed by [Zhou et al. \(2017\)](#) outperforms the ANNs based models. Furthermore, the extended variants of RNN, such as Long Short Term Memory networks (LSTM) and Gated Recurrent Unit (GRU), have been applied to car-following behavior modeling ([Huang et al., 2018; Wang et al., 2018; Wang et al., 2019](#)). The above neural network based car-following models not only used different network architectures but also had various model configurations such as input and output variables, and time steps. To be more specific, we will discuss the configurations in Section 3.3.

2.2. Reaction delay

Reaction delay is a common characteristic of humans in operation and control, such as driving a car, and it integrates mental processing time, movement time, and device response time ([Green, 2000](#)). It is reported that driver reaction delay affects the traffic dynamics in both microscopic and macroscopic way ([Aycin and Benekohal, 2004](#)). To better understand its impact on car-following, researchers conduct reaction delay estimation and incorporate it into car-following models, as reviewed below.

The reaction delay estimation is built on the relationship of stimulus-response, as the reaction delay defined as the time lag between the change in driving condition and the following response ([Gurusinge et al., 2002](#)). Two categories of estimation approaches are available in the literature. On one hand, the estimation is based on the observation of driver actions under either simulators or real environment ([Johansson and Rumar, 1971; Fambro et al., 1998; Green, 2000; Magister et al., 2005; Bilban et al., 2009](#)), which record the time lag between stimuli (for example, red traffic light and the expected and unexpected dangers) and driver reaction (such as braking the pedal). On the other hand, the approach that estimates reaction delay from real trajectory data is applied by many researchers. [Gazis et al. \(1961\)](#) first introduced cross-correlation method to identify the highest correlation between the relative speed profile and the following acceleration profile. [Triggs and Harris \(1982\)](#) obtained the reaction time distribution by measuring time gaps of the braking signal between the leading vehicle and the following vehicle. [Ozaki \(1993\)](#) proposed a piecewise linear function considering the asymmetric characteristic of acceleration and deceleration. [Ranjitkar et al. \(2003\)](#) developed a computer program to calculate the time interval between two neighboring peak points of relative speed and following acceleration profiles. [Ma and Andréasson \(2006\)](#) adopted spectrum analysis methods based on Fourier analysis of car-following data to estimate reaction delay. [Zhang and Bham \(2007\)](#) proposed a new graphical method for reaction time estimation under steady-state car-following condition. [Sharma et al. \(2019\)](#) detected the stimulus-response points under the traditional environment and the connected environment using wavelet-based energy distribution of time series of speeds. According to the available references, it is widely recognized that the reaction delay is highly related to the car-following trajectory profiles.

Reaction time was first introduced to car-following modeling by the GHR model. Since then, many researchers have incorporated reaction delay in mathematical car-following modeling ([Saifuzzaman and Zheng, 2014](#)), and reaction delay has become an indispensable human driving factor for the investigation of car-following behaviors. Reaction delay is taken into consideration in recent neural networks based car-following models. [Jia et al. \(2003\)](#) proposed an ANN car-following model predicting the acceleration at the last time step and explained this delay with drivers' reaction time. [Khodayari et al. \(2012\)](#) incorporated the estimated instantaneous reaction delay into the proposed ANN car-following model as one of the input variables. [Zheng et al. \(2013\)](#) further presented reaction delay estimation much explicitly with ANN method and proposed an ANN car-following model considering instantaneous reaction delay to predict car-following behavior in the next time step. Reaction delay was adopted to define the input time steps of deeper neural networks based car-following models, for example, the recurrent neural network car-following model proposed by [Zhou et al. \(2017\)](#), and the deep reinforcement learning car-following model proposed by [Zhu et al. \(2018\)](#). Although reaction delay is considered in neural network based car-following models, the above references show that its utilities vary across studies.

2.3. Data preparation

The NGSIM dataset ([FHWA, 2008](#)) is widely used to calibrate car-following models in previous studies. The NGSIM I-80 dataset was collected on the northbound direction of Interstate 80 (I-80) in Emeryville, California, on April 13, 2005. Traffic states of this dataset transform from uncongested flow to congestion, and a high degree of vehicle interaction exhibits as vehicles merging on and off the highway at the nearly-congested flow. This multiple phase property can provide higher flexibility for data-driven models that extract car-following behavior directly from mass field data.

The availability of high-fidelity trajectory data gives an opportunity to enhance the calibration of car-following models. However, as investigated by [Punzo et al. \(2011\)](#), the original NGSIM data set has measurement errors, and the errors have negative effects on the

performance of data-driven models. Montanino and Punzo (2015) proposed a “traffic-informed” method to ensure that the reconstructed trajectory data is consistent with vehicle kinematics and microscopic traffic dynamics and apply it to NGSIM reconstruction. Therefore, we adopt this reconstructed NGSIM I-80 dataset from 4:00 to 4:15 (Montanino and Punzo, 2013). The study area is approximately 400 m lengths with 5 regular lanes (from lane 2 to lane 6), excluding the high-occupancy vehicle lane. The data resolution is 10 Hz.

To eliminate the influence of lane-changing, we extract vehicle pairs whose car-following behaviors are continuously lasting no less than 30 s. Therefore, we have trajectory samples of 1386 vehicle pairs in total. Fig. 1 shows the comparison of raw NGSIM data and clean data from a random selected sample.

Vehicles from one lane can be selected to compose the test dataset due to the need for complete independence of the testing process. Therefore, we select a total of 332 vehicle pairs (167975 trajectory samples) collected from lane 2nd as the test dataset. The trajectory samples extracted from other lanes (from lane 3rd to lane 6th) are applied to model calibration, for a total of 494403 trajectory samples from 1054 vehicle pairs.

3. Methodology

3.1. Long Short Term Memory (LSTM) networks

Recurrent neural networks (RNNs) are a type of neural network where the outputs from previous time steps are taken as inputs for the current time step. However, conventional RNNs have limitations for learning long-term dependencies due to vanishing gradient (Bengio et al., 1994). Long Short Term Memory (LSTM) is specifically designed to address long-term dependency problems and can carry information across multiple time steps.

Fig. 2 shows how the LSTM unit keeps updating memory by controlling information flow. Where x_t denotes the input vectors. c_t and h_t denote the memory cell state and output the LSTM unit at time t . f_t , i_t and o_t are the gating vectors. The sigmoid function $\sigma(\cdot)$ and the hyperbolic tangent function $\tanh(\cdot)$ are both activation functions, as defined in Eqs. (1) and (2), which convert the values in the range of $(0, 1)$ and $[-1, 1]$.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2)$$

Corresponding to the three controlling gates (forget gate, input gate, and output gate) in Fig. 2, Eqs. (3)–(8) illustrate the calculation process step by step. Where W_{fh} , W_{fx} , W_{ih} , W_{ix} , W_{ch} , W_{cx} , W_{oh} , and W_{ox} are weighted matrices, and b_f , b_i , b_c , and b_o are the bias in gates.

Forget gate:

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \quad (3)$$

This gate gets information from the input of the current step x_t as well as the output of the previous step h_{t-1} . The sigmoid activation function deals with this combined information to decide how much information can be thrown away from the cell state.

Input gate:

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \quad (4)$$

$$\tilde{c}_t = \tanh(W_{ch}h_{t-1} + W_{cx}x_t + b_c) \quad (5)$$

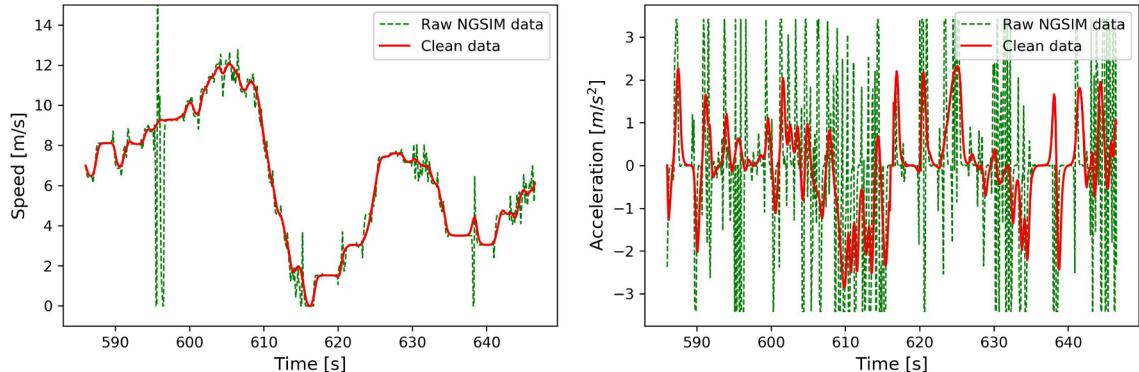


Fig. 1. Comparison of raw and clean data for a sample trajectory (vehicle 1929).

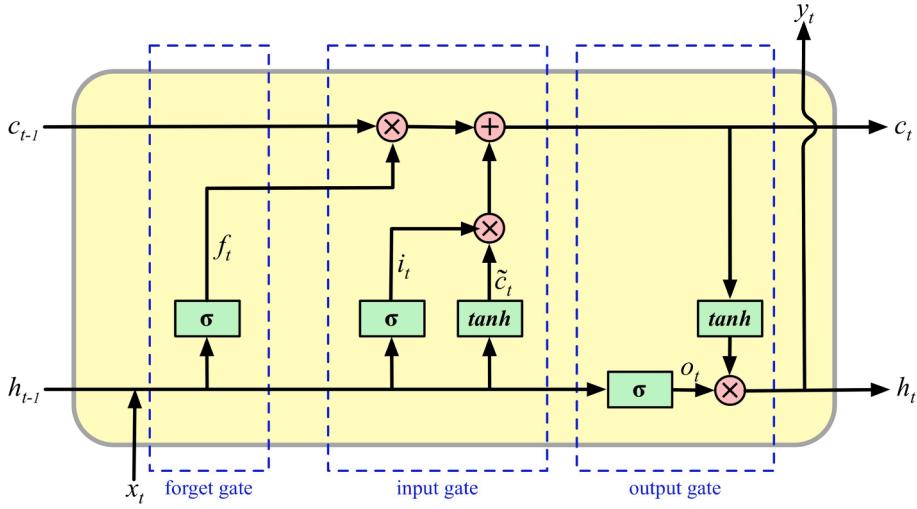


Fig. 2. Internal structure of LSTM unit.

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \quad (6)$$

This gate filter the combination information and store in an input vector i_t . The candidate \tilde{c}_t is calculated for updating the memory cell. Then, the previous memory cell c_{t-1} is multiplied by f_t and added to the filtered candidate $i_t \otimes \tilde{c}_t$ to obtain the current cell state c_t .

Output gate:

$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \quad (7)$$

$$y_t = h_t = o_t \otimes \tanh(c_t) \quad (8)$$

In this final gate, the output y_t is also the hidden state h_t that can be calculated from the element-wise multiplication of filter o_t and the current cell state c_t .

LSTM has achieved excellent performance on sequence forecasting problems (Hochreiter and Schmidhuber, 1997). It has been applied in many aspects of transportation research (Nguyen et al., 2018), such as day-to-day travel demand forecasting (Cheng et al., 2016), traffic conditions modeling (Fouladgar et al., 2017), short-term traffic forecasting (Zhao et al., 2017), and car-following modeling (Huang et al., 2018). However, for various applications, the topology of LSTM is no different from the typical RNN architecture. There are two most commonly used architectures of LSTM models, as shown in Fig. 3. One common type is the many-to-one architecture which means that it can comprehend all the input and has one output y_t at the last time step, as shown in Fig. 3(a). And Fig. 3(b) represents the other common type, the many-to-many architecture, where the length of the input sequence and the length of the output sequence are identical.

Similar to the natural language processing problems, many time series prediction problems probably have the length of the output sequence greater than one and different from the length of the input sequence. With special values padding and masking, the input and output sequences can have equal lengths and fit in the LSTM architectures. However, this is not flexible enough. Therefore, a more

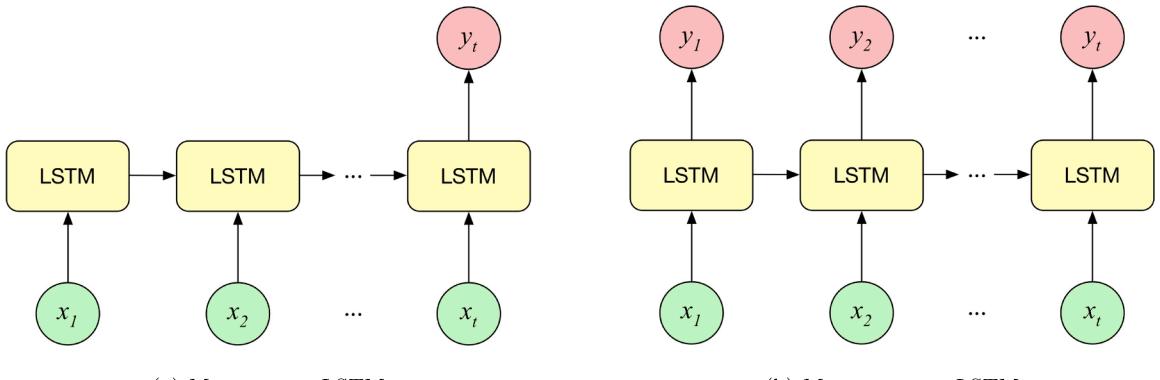


Fig. 3. LSTM architectures.

flexible architecture that can handle any length of input and output sequence is needed.

3.2. Sequence to Sequence (seq2seq) car-following model

Cho et al. (Cho et al., 2014) proposed an approach which combines the Encoder-Decoder architecture with RNN to solve the phrase representations learning problem, which was the very first sequence to sequence (seq2seq) model. Additionally, Sutskever et al. (2014) achieved state-of-the-art results in the field of machine translation with a refined seq2seq model, namely Encoder-Decoder LSTM architecture. In recent years, this architecture has received much attention (LeCun et al., 2015). It is skilled in solving the many-to-many time series prediction problems, especially when the length of the input sequence and output sequence are different, which is suitable for car-following behavior modeling.

We adopt the seq2seq architecture to predict car-following behaviors. This architecture consists of two LSTM networks (one for encoding and the other for decoding) which support variable input sequence length (T) and output sequence length (L), as illustrated in Fig. 4. Corresponding to the illustration, the formulas are shown in Eqs. 9–12.

Encoder:

$$h_t = f(X_t, h_{t-1}, c_{t-1}) \quad (9)$$

$$C = h_T \quad (10)$$

In this part, the LSTM units first take in the information from input sequence $X = (X_1, \dots, X_T)$ and encode it step by step. Where $f(\cdot)$ denotes the function of LSTM unit. c_t and h_t denote the memory cell state and endogenous output of LSTM unit at step t . Then, the last step output h_T passes into a context vector C , which represents the summary information of the input sequence.

Decoder:

$$p(Y_t | \{Y_1, \dots, Y_{t-1}\}, C) = g(Y_{t-1}, h_t, C) \quad (11)$$

$$p(Y) = \prod_{t=1}^L p\left(Y_t \middle| \{Y_1, \dots, Y_{t-1}\}, C\right) \quad (12)$$

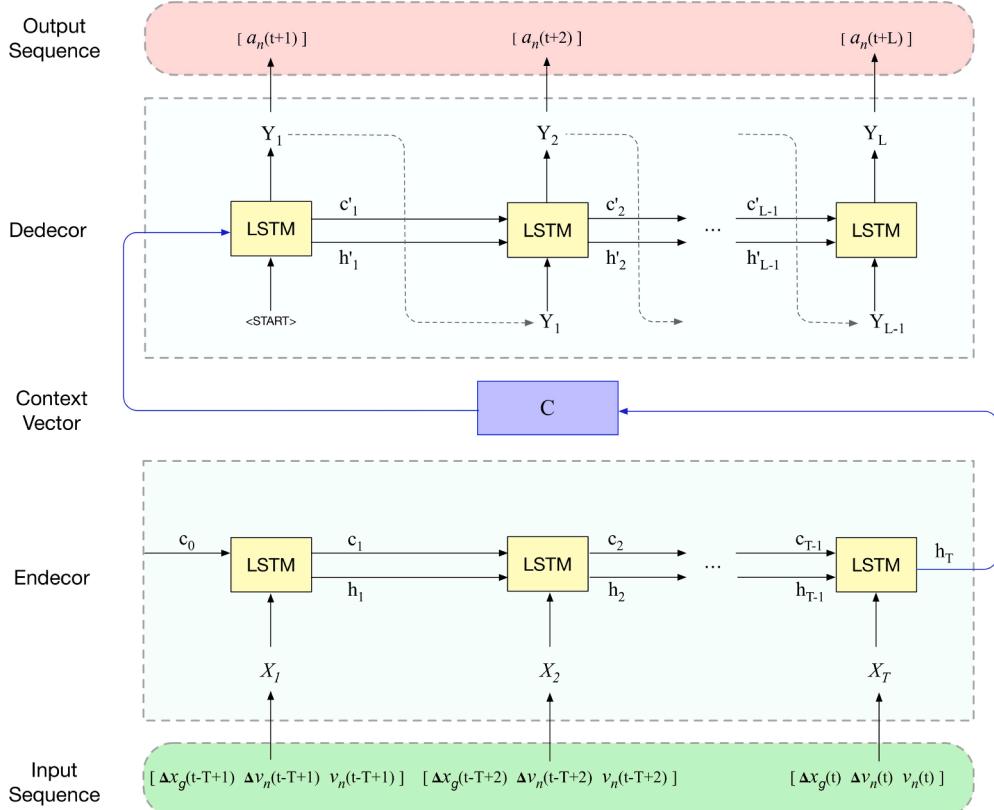


Fig. 4. The framework of seq2seq car-following model.

In the process of decoding, the context vector C is used as the initial hidden state of the decoder network, and the memory cell state (c'_t) and the endogenous output (h'_t) at time step t are fed into the next LSTM unit. Where $g(\cdot)$ denotes the function of LSTM unit. The seq2seq model is trained to maximize the conditional probability of the output sequence $p(Y)$.

3.3. Configurations of the seq2seq car-following model

Since traffic characteristics may vary with different traffic conditions such as expressway and urban street, the framework must be modified to accommodate local traffic conditions so that the model can simulate car-following behaviors accurately. A summary of the previous representative configurations of neural network based car-following models can be found in Table 1. The seq2seq car-following model is configured based on the analyses of these representative configurations and the 1054 vehicle pairs' sampled data.

3.3.1. Input and output variables

The subject vehicle (n) adjusts its speed according to the state of the leading vehicle ($n-1$) during a car-following situation. The mathematical car-following models establish equations to describe this movement. With regards to neural networks based car-following models, space headway (Δx_h), gap distance (Δx_n), relative speed difference (Δv_n) and speed of the subject vehicle (v_n) are the most commonly used input variables, as presented in Table 1. These four variables have decisive effects on the action of the subject vehicle. However, the gap distance is more reasonable than the space headway, because the vehicles' lengths differ from each other, especially the heavy vehicles. Moreover, the observation of the space from the subject vehicle's front bumper to the leading vehicle's rear bumper is straightforward for both human-driven vehicles (HDVs) and connected and autonomous vehicles (CAVs). For the output variable, it is conventional to use the speed (v_n) or acceleration (a_n) of the subject vehicle, as shown in Table 1. In the view of vehicle controlling, a_n is an effective prediction of vehicle's movement for the next time steps and more direct than v_n . Therefore, we adopt Δx_n , Δv_n and v_n as the input variables and a_n as the output variable for the seq2seq model.

Therefore, the input sequence and output sequence in the seq2seq model are settled. As the framework shown in Fig. 5, t is the current time step, and the input sequence is a matrix of three input variables for T time steps (from $t-T+1$ to t). The input sequence is fed into the seq2seq model and makes multi-step predictions, where the output sequence is a matrix of acceleration for L time steps (from $t+1$ to $t+L$). The mapping function is formulated as Eq. (13):

$$a_n(t+1), \dots, a_n(t+L) = Fun(\Delta x_n(t-T+1), \dots, \Delta x_n(t), \\ \Delta v_n(t-T+1), \dots, \Delta v_n(t), \\ v_n(t-T+1), \dots, v_n(t)) \quad (13)$$

Where $Fun(\cdot)$ denotes the mapping function between input and output sequence, which will be learned by Encoder-Decoder LSTM architecture.

3.3.2. Historical time steps based on memory effect

Lee (1966) first introduced driving memory into the GHR model. And, Toledo (2007) has pointed out that the driver's decision-making is always dependent on the historical driving behaviors and the past traffic states. Many researchers tried to embed the memory effect into mathematical car-following behavior modeling (Treiber and Helbing, 2003; Tang et al., 2009; Sipahi and Niculescu, 2010; Yu and Shi, 2015; Liu et al., 2016) and demonstrated that car-following models with the memory effect taken into account

Table 1
Representative configurations of neural network based car-following models.

Researchers	Architecture	Data resolution	Input variables	Historical time/time step	Output variables	Prediction horizons/time step
Jia et al. (2003)	ANN	2 Hz	$\Delta x_h, \Delta v_n, v_n$, desired speed	0.5s/ 1	a_n	0.5s/ 1
Panwai and Dia (2007)	ANN	10 Hz	$\Delta x_h, v_{n-1}$	0.1s/ 1	v_n	0.1s/ 1
Chong et al. (2011)	ANN	10 Hz	$\Delta x_h, \Delta v_n, v_n$	0.1s/ 1	a_n	0.1s/ 1
Khodayari et al. (2012)	ANN	1 Hz	$\Delta x_h, \Delta v_n, v_n$, instantaneous reaction delay	1s/ 1	a_n	1s/ 1
Zheng et al. (2013)	ANN	1 Hz	$\Delta x_h, \Delta v_n, v_n$	1s/ 1	v_n	1s/ 1
Zhou et al. (2017)	RNN	10 Hz	$\Delta x_h, \Delta v_n, v_n$	0.1s/ 1	a_n	0.1s/ 1
Morton et al. (2017)	LSTM	10 Hz	$\Delta x_h, \Delta v_n, v_n, a_n$	0.1s/ 1	distribution of a_n	0.1s/ 1
Huang et al. (2018)	LSTM	10 Hz	$\Delta x_h, \Delta v_n, v_n$	5s/ 50	v_n	0.1s/ 1
Wang et al. (2018), Wang et al. (2019)	GRU	1 Hz	$\Delta x_h, \Delta v_n, v_n$	10s/ 10	v_n	1s/ 1

Δx_h is the space headway (from the leading vehicle's front bumper to the subject vehicle's front bumper).

Δx_n is the gap distance (from the leading vehicle's rear bumper to the subject vehicle's front bumper).

Δv_n is the relative speed difference between the subject vehicle n and the leading vehicle $n-1$.

v_n and v_{n-1} are the speed of the subject vehicle n and the leading vehicle $n-1$.

a_n is the acceleration of the subject vehicle n .

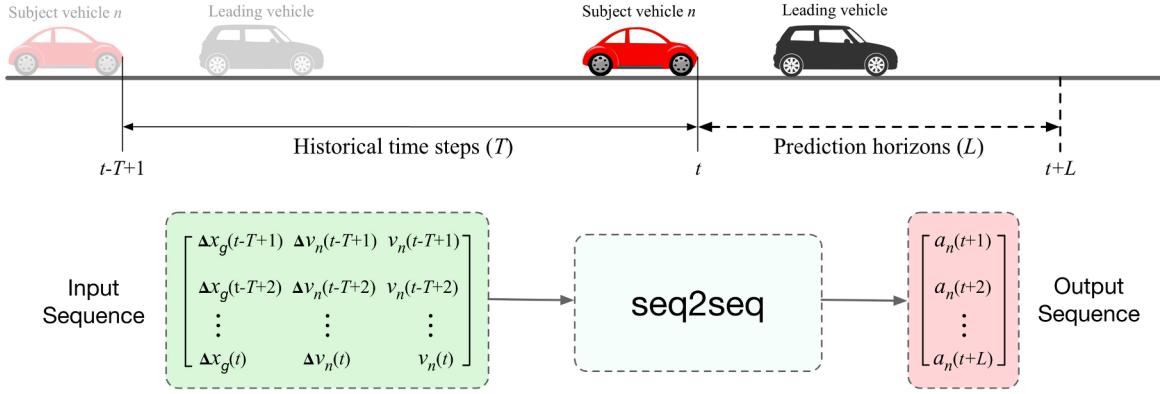


Fig. 5. Framework overview.

could better describe the traffic flow characteristics. However, it is difficult for mathematical car-following models to consider the long-time period memory effect. Therefore, incorporating memory effect into data-driven car-following models has received attention, due to its advantages of handling massive historical information.

Memory effect has been incorporated into RNNs car-following models, and it is demonstrated that memory effect benefits the prediction of car-following behavior. Wang et al. (2018) introduced the memory effect into a deep learning car-following model by defining the historical time of input variables and setting the time steps as 10 ($T = 10$), as shown in Table 1. Furthermore, Wang et al. (2019) demonstrated that long memory is important and should be embedded into data-driven car-following models, by studying the relationship between memory effect and hysteresis phenomena in congested freeway traffic. Huang et al. (2018) introduced that LSTM-NN has an advantage of incorporating memory effect to capture realistic traffic flow characteristics, and the forget gate in LSTM unit can not only capture historical information but also gradually remove the memory when the historical information is out of date. Moreover, they proposed a LSTM-NN car-following model with 50 historical time steps ($T = 50$), as shown in Table 1.

The LSTM unit in our seq2seq car-following model can deal with time-sequence historical information, which is consistent with the human driving process that drivers make decisions with varying short-term memory. The length of historical time steps (T) can be determined by comparing the testing results, and $T = 50$ (5 s) is long enough for our seq2seq model. The detailed testing process is provided in Section 4.1.

3.3.3. Prediction horizons based on reaction delay

As the reaction delay studied in previous literature (Gazis et al., 1961; Gurusinghe et al., 2002; Ranjikar et al., 2003; Ma and Andréasson, 2006; Zheng et al., 2013; Sharma et al., 2019; Zhu et al., 2020), it is widely recognized that the reaction delay can be defined based on the stimulus–response concept of car-following that first put forward by Chandler et al. (1958), as shown in Eq. (14).

$$a_n(t + \tau_0) = \lambda[v_{n-1}(t) - v_n(t)] \quad (14)$$

According to this concept, the relative speed ($v_{n-1}(t) - v_n(t)$) is regarded as the stimulus and the acceleration ($a_n(t + \tau_0)$) as the response. τ_0 is the reaction delay that can be defined by the time lag between the neighboring peaks of relative speed and acceleration, as the arrows shown in Fig. 6. However, it is difficult to find each time lag manually. Thus, we formulate the correlation method suggested by Gazis et al. (1961) in Eqs. (15)–(17) to find the time lags between relative speed and acceleration series. With regards to the testing time lag τ that ranges from α to β , the cross-correlation coefficient $\rho_{\Delta v a}(\tau)$ is calculated and compared. Then, the estimated reaction delay is defined by the time lag τ^* that has the highest cross-correlation coefficient $\rho_{\Delta v a}(\tau^*)$. This estimating process is implemented in python with sampled data.

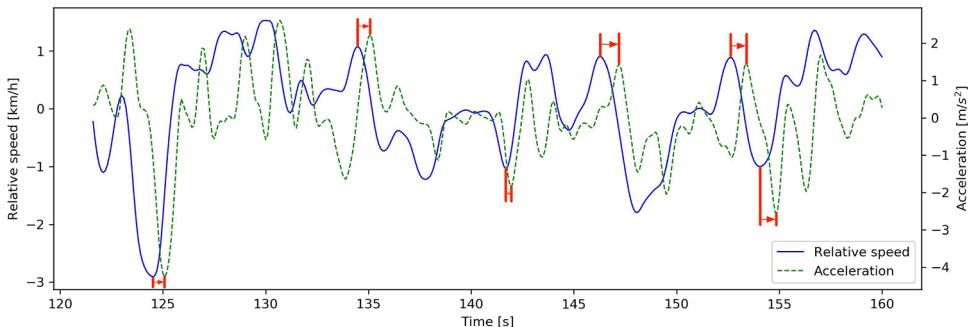


Fig. 6. The profiles of relative speed and acceleration.

$$R_{\Delta v a}(\tau) = E[\Delta v(t)a(t+\tau)] \quad (15)$$

$$\rho_{\Delta v a}(\tau) = \frac{R_{\Delta v a}(\tau) - \mu_{\Delta v} \mu_a}{\gamma_{\Delta v} \gamma_a} \quad (16)$$

$$\tau^* = \{\tau | \max(\rho_{\Delta v a}(\tau)), \alpha \leq \tau \leq \beta\} \quad (17)$$

Where $R_{\Delta v a}(\tau)$ denotes the cross-correlation function. $E[\cdot]$ is the expectation function. $\Delta v(t)$ and $a(t+\tau)$ denote the relative speed and acceleration series. $\mu_{\Delta v}$ and μ_a are the mean value of series. $\gamma_{\Delta v}$ and γ_a are the standard deviation of series. α and β are the minima and maxima of testing time lags.

To obtain reasonable minima and maxima values for testing, we define the range of reaction delay carefully based on the available references. Researchers have estimated reaction delay with different methods and based on different datasets, and the results are not quite agreed, as summarized in Fig. 7(a). Papathanasopoulou and Antoniou (2015) have defined the reaction delay in a wide range from 0.4 s to 3.0 s by considering the previous studies. Moreover, Yeo (2008) has confirmed that the minimum value of reaction delay is around 0.5 s. Therefore, it is reasonable to set the range of reaction delay from 0.4 s to 3.0 s, namely $\alpha = 4$ and $\beta = 30$.

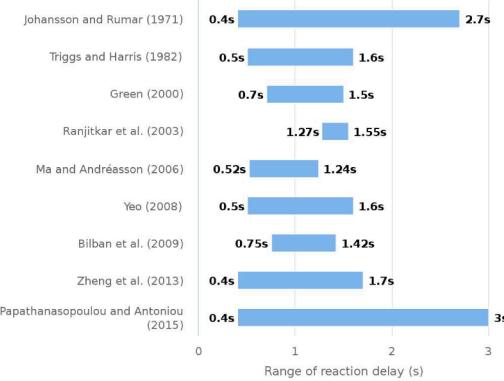
To capture the time-varying property of reaction delay, we separate each vehicle trajectory into multiple time intervals and estimate their reaction delay individually. The length of the trajectories subsection can be set as 5 s, 8 s, and 10 s for testing, and calculation result shows that their mean values of estimated reaction delay are 1.33 s, 1.03 s, and 0.98 s, respectively. To the best of our knowledge, Zhang and Bham, 2007; Zheng et al., 2013 are the few references that use the NGSIM dataset to estimate reaction delay, and the mean values of their estimation are 0.86 s and 0.98 s. We compare these mean values and find that the estimation from the length 10 s shows a good match with the references. This comparison is rather arbitrary, but it is useful to identify the main trend of the estimation results. Therefore, the subsection length is chosen as 10 s, and 4313 estimated reaction delay values are obtained from 1054 vehicle pairs. The graphical and statistical descriptions of the estimated reaction delay are shown in Fig. 7(b) and Table 2.

The existing neural networks based car-following models predict the output variable in the next time step, as the prediction horizons presented in Table 1. However, driving action, in reality, is sequential in the next several time steps rather than being made step by step. Moreover, the traffic state in the current time step may not fully reflect in the next time step because of the reaction delay. Therefore, we propose multi-step predictions with our seq2seq model. The length of prediction horizons (L) is assumed based on the statistic description of estimated reaction delay (τ^*) in Table 2. To find the best prediction horizons, we test $L = 1, 4, 6, 8, 10, 12, 15, 20, 30$ one by one, as the detailed testing results given in Section 4.1. By comparing the results, $L = 12$ ($\tau^* = 1.2$ s) is defined as the prediction horizons of our seq2seq model.

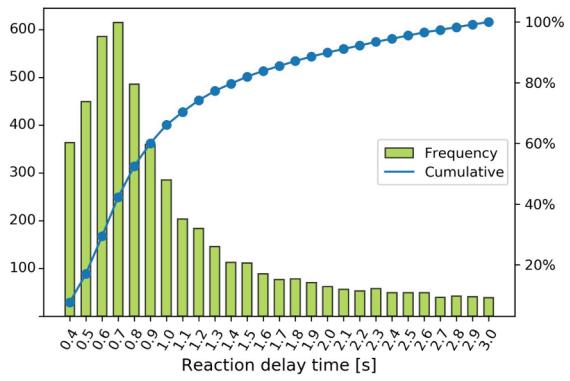
4. Experiments

4.1. Calibration of models

The seq2seq car-following model is calibrated on the trajectory samples from 1054 vehicle pairs. In this process, the training dataset is randomly selected, taking up the samples of 70% vehicle pairs, and the samples from the remaining 30% pairs are collected as validation dataset. The model is trained only on the training set, and the performance on the validation set is evaluated by loss function at the end of every epoch. The number of epochs controls the number times that the learning algorithm passes through the entire training dataset. In each epoch, every sample in the training dataset has an opportunity to contribute to the updating of internal model parameters. Besides, the “Early Stopping” mechanism is introduced to protect the model from overfitting by stopping the training process when the validation loss does not decrease for 5 consecutive epochs.



(a) Range of reaction delay according to references



(b) Histogram and cumulative curve of reaction delay

Fig. 7. The estimation for reaction delay.

Table 2

Statistical description of estimated reaction delay.

Count	Mean (SD)	Min	Max	Percentile [25%, 50%, 75%]
4313	0.98 (0.64)	0.4	3.0	[0.6, 0.8, 1.2]

The adaptive optimizer Adam (Kingma and Ba, 2014) is adopted as the optimization algorithm, and the parameters in Adam are defined as follows: $lr = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-08$, $decay = 0.0$. The hyperbolic tangent function $tanh(\cdot)$ is chosen as the activation function in both the Encoder part and the Decoder part. Through experimental optimization, the best choices of other hyper-parameters are presented as follows:

- The loss function: mean squared error
- The number of neurons in LSTM unit: 32
- The depth of Encoder-Decoder: 1 (without LSTM stack)
- Batch size: 128

Different historical time steps ($T = 10, 30, 50, 80, 100$) and prediction horizons ($L = 1, 4, 6, 8, 10, 12, 15, 20, 30$) are tested for the seq2seq model based on above hyper-parameters. Mean Square Error (MSE) between the predicted value and the actual value is adopted as the performance index, and the performance comparison for different time steps are presented in Fig. 8. Where $T = 50$ and $L = 12$ achieve the smallest MSE value and indicate the best performance. Besides, the MSE value of 1-time step, as shown in Fig. 8(b), indicates that the model considering no reaction delay has poor performance. Therefore, the best choices of historical time steps and prediction horizons are 50 and 12.

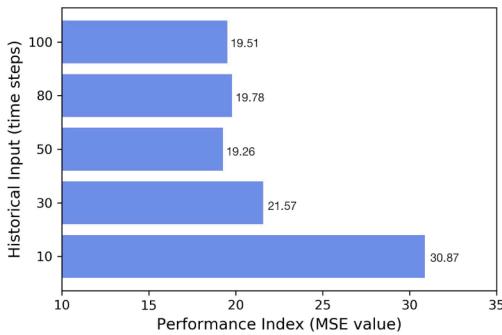
The seq2seq model is compared with the LSTM model and a representative mathematical model (IDM) in terms of simulation results. The LSTM model has the same configuration and hyper-parameters as the seq2seq model, such as depth (without LSTM stack) and neurons' number (32). Besides, the IDM (Treiber et al., 2000) is formulated by Eqs. (18) and (19).

$$a_n(t) = \tilde{a} \left[1 - \left(\frac{v_n(t)}{\tilde{v}} \right)^4 - \left(\frac{S(v_n(t), \Delta v_n(t))}{\Delta x_n(t)} \right)^2 \right] \quad (18)$$

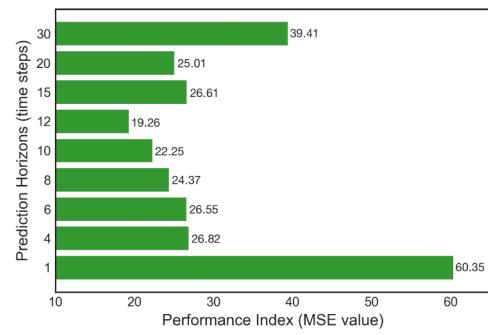
$$S(v_n(t), \Delta v_n(t)) = s_0 + t_0 v_n(t) - \frac{v_n(t) \Delta v_n(t)}{2\sqrt{\tilde{a}\tilde{b}}} \quad (19)$$

Where $S(v_n(t), \Delta v_n(t))$ is the desired space headway function and calculated from the speed (v_n) and the relative speed (Δv_n). The desired speed (\tilde{v}), the maximum acceleration (\tilde{a}), the maximum deceleration (\tilde{b}), the desired time headway (t_0) and the minimum space headway (s_0) need to be calibrated.

In terms of the IDM calibration, Zhou et al. (2017) have achieved outstanding simulation accuracy with their parameters calibrated by the Global Least-Squared Errors method (Treiber and Kesting, 2013) at trajectory level. The trajectory data used in our paper and Zhou et al. (2017) represent the same characteristics of traffic dynamics, due to they are both extracted from the NGSIM I-80. And, the simulation accuracy is both quantified by the Mean Squared Error (MSE) between actual and predicted data. Therefore, we use these calibrated parameters for the IDM, as listed in Table 3.



(a) Comparison of different historical time steps



(b) Comparison of different prediction horizons

Fig. 8. The experimental test for the configuration of historical time steps and prediction horizons.

4.2. Car-following models comparison

The seq2seq car-following model is applied to trajectory simulation. The simulated trajectory can be calculated with the discrete-time car-following process based on the prediction of acceleration, as formulated in Eq. (20). And, the Mean Squared Error (MSE) of simulated trajectory and observed trajectory is calculated with Eq. (21), which is used to evaluate the performance.

$$\begin{cases} \hat{v}_n(t+1) = \hat{v}_n(t) + \hat{a}_n(t+1)\Delta t \\ \hat{x}_n(t+1) = \hat{x}_n(t) + \hat{v}_n(t)\Delta t + \frac{1}{2}\hat{a}_n(t+1)\Delta t^2 \end{cases} \quad (20)$$

$$\text{MSE} = \frac{1}{M} \sum_{i=1}^M [x_n(i) - \hat{x}_n(i)]^2 \quad (21)$$

Where \hat{a}_n denotes the predicted acceleration. \hat{v}_n and \hat{x}_n denote the estimated speed and location. x_n denotes the observed location.

The trajectory simulation on the training and validation dataset (1054 vehicle pairs) and on the test dataset (332 vehicle pairs), are conducted with the seq2seq model, and the evaluation results are compared in Fig. 9. The model shows better performance on the training and validation dataset than on the testing dataset, which indicates it has a good fitting quality with known behavior.

To verify the performance of the car-following models, we simulate trajectories on the test dataset (332 vehicle pairs) with the other two models (IDM and LSTM) for comparison. The simulation results are evaluated by the performance index MSE, and their MSE values are distributed with different Mean values (28.59 for IDM, 29.29 for LSTM, and 19.26 for seq2seq), as shown in Fig. 10. To illustrate the trajectories comparison, we select Vehicle 1898 and plot the observed and simulated trajectory profiles in Fig. 11, because the simulated trajectories of Vehicle 1898 have MSE values approximate to the mean values and can better represent the models' performance. It shows that all the models can follow the trend of observed trajectory well, whereas the simulated trajectory of the seq2seq model is more consistent with the field data.

To make a more general conclusion, we compare the models' performance on the test dataset by statistical analysis, as shown in Table 4. IDM is an outperforming mathematical car-following model, and the simulation demonstrates that it can give reliable results for trajectory reproducing. The simulation result of LSTM even turns out to be a little worse than IDM, due to the reason for a fair comparison that we have restricted its configuration to the same as the seq2seq model (such as 32 neurons in the LSTM unit and without LSTM stack). It has been demonstrated that LSTM can achieve better performance with a more complex configuration (Huang et al., 2018). However, the mean and standard deviation of MSE from the seq2seq model, 19.26 and 26.72 respectively, are much smaller than that from the other two models. Moreover, other statistics indexes such as the percentiles show that the MSE values of seq2seq model have a smaller range and are more densely distributed. The results indicate that the seq2seq model not only yields higher accuracy for reproducing trajectories but also shows a more stable predictive quality.

Furthermore, to better explore the models' performance, we investigate the simulated trajectories and estimate the models' performance of capturing heterogeneous driving behaviors. Researchers have incorporated heterogeneous driving behaviors into mathematical car-following models (Laval and Leclercq, 2010; Chen et al., 2012a; Peng et al., 2016) as well as analyze the heterogeneity with data-driven car-following models (Zhou et al., 2017). The heterogeneous driving behaviors can be divided into three types, including aggressive behavior, timid behavior, and normal behavior. The driver who has aggressive driving behavior often anticipates future traffic conditions in advance and keeps a small gap distance with the leading vehicle, and the timid drivers tend to drive carefully and keep a large gap. While normal driving behavior shows up with a relatively moderate gap. We calculate the global gap distance of every vehicle pair on our test dataset, and the results show that drivers keep the gaps from 5.15 m to 30.71 m with an average of 13.11 m. According to these gap distances, we distinguish vehicle trajectories into the three driving behavior types, as the representative vehicle profiles shown in Fig. 12. In terms of heterogeneous driving behaviors, car-following models have different performance levels. As shown in Fig. 12(a), the trajectory profiles for normal driving behavior predicted by IDM, LSTM, and seq2seq present comparable accuracy. However, the LSTM and seq2seq models have much stronger performance than the IDM for aggressive and timid driving behaviors (Fig. 12(b) and (c)). It indicates that deep learning car-following models can capture the driving behavior heterogeneity by learning the hidden information from the initial trajectory. Besides, the seq2seq model performs better than the same depth LSTM model under all situations, which implies the effectiveness and superiority of the seq2seq model on predicting driving behavior.

4.3. Platoon simulation

Platoon simulation is an important application of car-following models because it can reproduce traffic phenomena such as oscillation and hysteresis. Platoon simulation means that except for the first following vehicle in a platoon is simulated by its initial

Table 3

Calibrated parameters of the IDM (Zhou et al., 2017).

IDM	\tilde{v}	\tilde{a}	\tilde{b}	t_0	s_0
Calibrated value	27.19	2.01	1.77	1.53	6.73

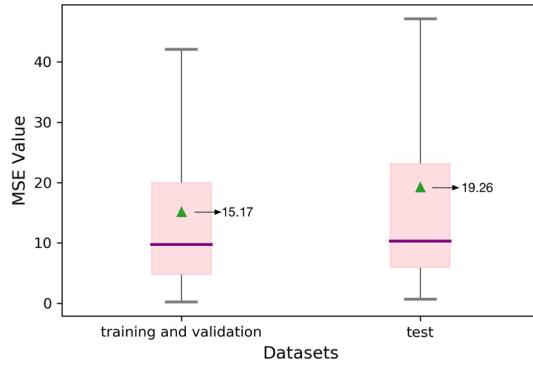


Fig. 9. MSE distribution in terms of datasets.

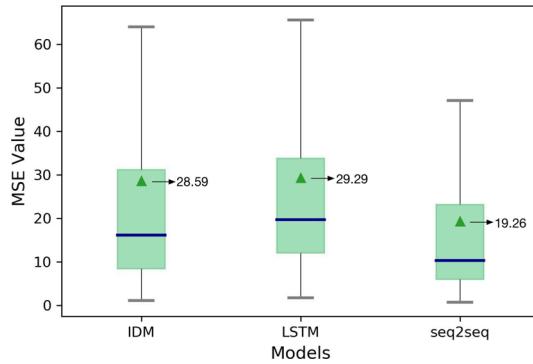


Fig. 10. MSE distributions in terms of models.

states and observed platoon leader, other subject vehicles in the platoon are simulated by their initial states and the simulated results of the vehicle ahead. To further validate the seq2seq car-following model's performance on reproducing traffic oscillation and hysteresis, we employ two platoons that traverse stop-and-go waves for platoon simulation, and the leaders are Vehicle 898 and Vehicle 2774. The time-space diagrams of the real and simulated trajectories are plotted in Fig. 13 and 14. The average MSE values are 41.51 and 90.78 for simulated platoon following Vehicle 898 and Vehicle 2774 respectively, which indicates the performance of platoon simulation is worse than that of simulated vehicle pairs because of accumulative error. The perturbation waves in both simulated platoons propagate from downstream to upstream show typical traffic oscillations.

The hysteresis loop in oscillation was first observed by Treiterer and Myers (1974) in the fundamental diagram. Then, Laval (2011) improved the hysteresis measurement method with Edie's generalized definitions (Edie, 1963). We use this method to aggregate density (k), flow (q), and speed (v) inside relatively stationary conditions, as formulated in Eqs. (22)–(24).

$$k = \frac{\sum_{i=1}^n t_i}{|A|} \quad (22)$$

$$q = \frac{\sum_{i=1}^n x_i}{|A|} \quad (23)$$

$$v = \frac{q}{k} = \frac{\sum_{i=1}^n x_i}{\sum_{i=1}^n t_i} \quad (24)$$

Where A is a parallelogram region containing n vehicles, with two opposite sides with slopes wave speed and the other two with a slope comparable to the vehicles' speed, and $|A|$ is the area of A . t_i and x_i denote the travel time and distance of the i th vehicle in area A .

The estimated flow-density evolution is calculated with Edie's method based on the corresponding regions in the time-space diagrams. For the platoon following Vehicle 898 (see Fig. 13), the flow-density plane exhibits two phases as vehicles traverse stop-and-go waves. One is a lower phase when vehicles decelerate, and the other is an upper phase when vehicles accelerate, which consequently

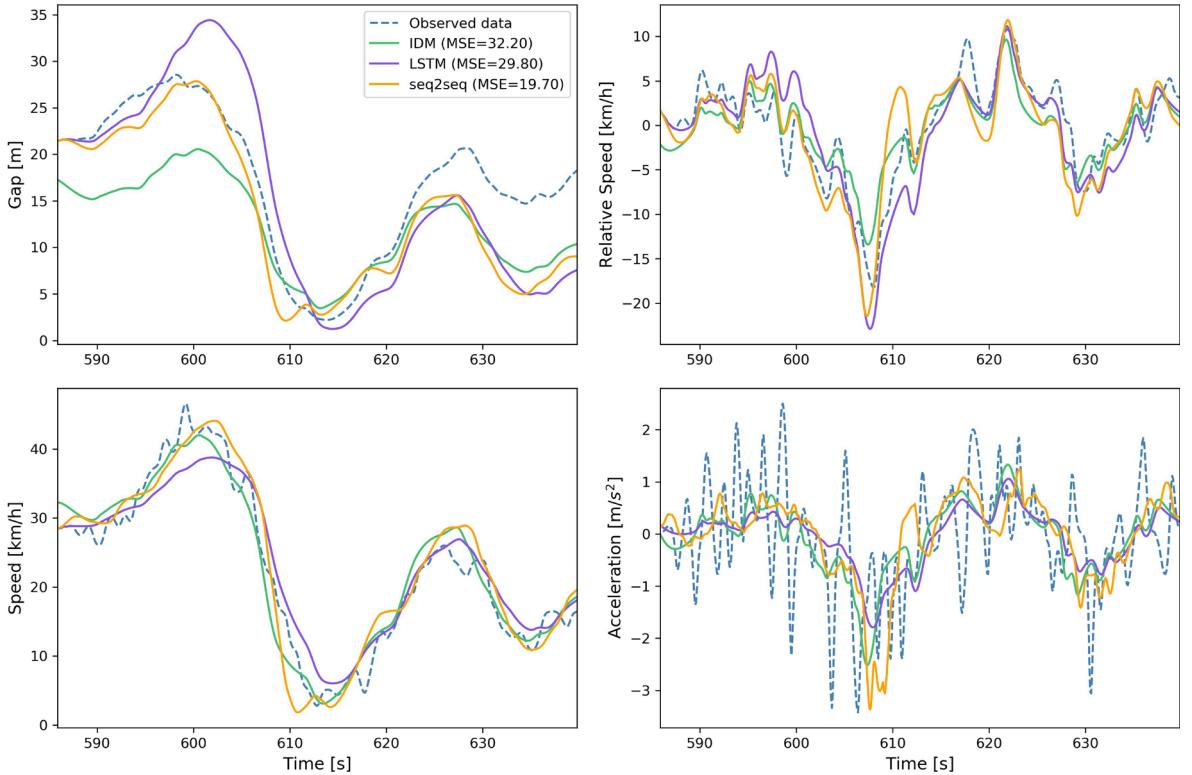


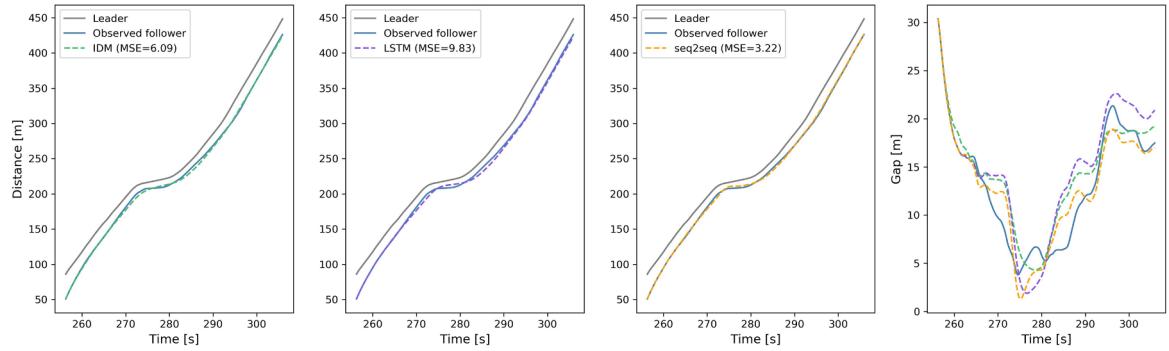
Fig. 11. Trajectory profiles of Vehicle 1898.

Table 4
Statistics results of the performance.

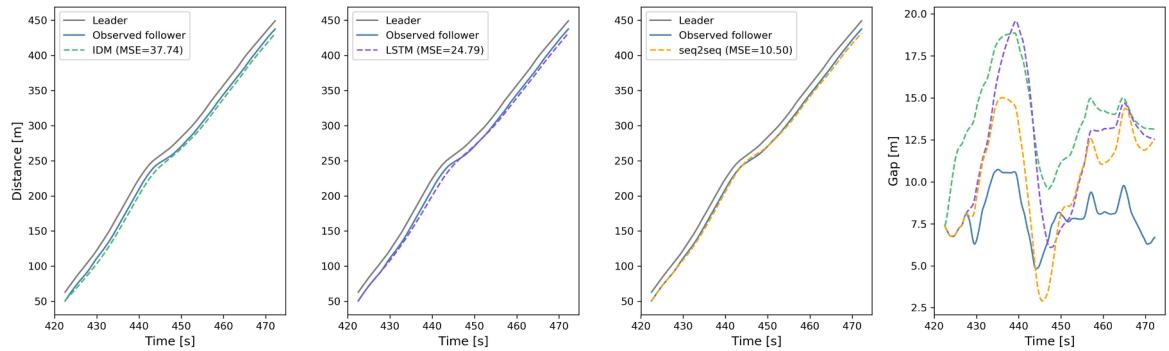
Model	MSE			
	Mean (SD)	Min	Max	Percentile [25%, 50%, 75%]
IDM	28.59 (56.79)	1.05	870.61	[8.38, 16.13, 31.13]
LSTM	29.29 (36.36)	1.67	451.73	[12.03, 19.67, 33.77]
seq2seq	19.26 (26.72)	0.69	209.77	[5.95, 10.29, 23.10]

form an anti-clockwise hysteresis loop. For the platoon following Vehicle 2774 (see Fig. 14), the deceleration phase is on top of the acceleration phase during flow-density evolution, which leads to a clockwise hysteresis loop. For both tested platoons, the orientations of hysteresis loops obtained from simulated trajectories are consistent with the observation. The clockwise and anti-clockwise loops for hysteresis phenomenon have been confirmed by Ahn et al. (2013), Chen et al. (2012), and they also correspond to the positive and negative hysteresis in Laval (2011), Wang et al. (2019). It is suggested by Chen et al. (2012b, 2012a, 2014), Laval (2011), Laval and Leclercq (2010) that aggressive and timid driver behaviors are the reasons for different hysteresis levels, and the relative position of acceleration and deceleration phases depends on the prevailing driving behavior inside the platoon. As the hysteresis phenomenon well reproduced in the simulated platoon, it further indicates that the seq2seq car-following model can capture driving behavior from real traffic.

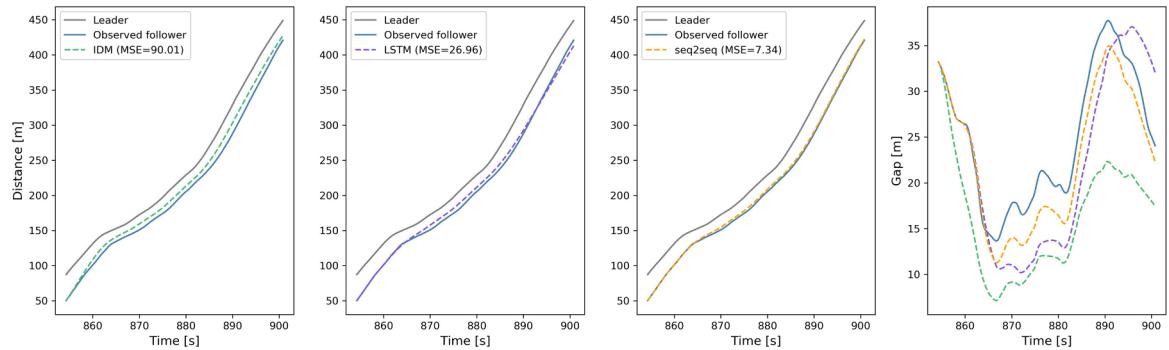
From the simulated trajectories in Fig. 13 and 14, it seems that the traffic oscillations are exaggerated by seq2seq car-following model. Treiber et al. (2006) have pointed out that the stability of traffic flow can be enhanced by spatial anticipation namely considering several vehicles ahead instead of only the immediate vehicle in front. And this spatial anticipation concept has been incorporated into mathematical car-following models (Lenz et al., 1999; Hoogendoorn et al., 2006; Treiber et al., 2006; Zhang, 2014; Yu and Shi, 2015; Zhou et al., 2016). In the data-driven modeling field, Lin et al. (2019) analyzed the spatial error propagation problem and applied a platoon-level training strategy with Int-LSTM model. In light of these studies, we extend the seq2seq car-following model with spatial anticipation to smooth traffic oscillations. n_a is the number of anticipated vehicles, and $n_a = 1$ corresponds to the original model without spatial anticipation considering only the nearest preceding vehicle. We incorporate the spatial anticipation into the seq2seq car-following model by taking the gap distance ($\Delta x_n^{n-\theta}$) and relative speed ($\Delta v_n^{n-\theta}$) between the subject vehicle n and multiple preceding vehicles $n-\theta$ ($\theta = 1, 2, \dots, n_a$) as input variables, as formulated in Eq. (25).



(a) Normal driving behavior: Vehicle 905 (average gap distance 13.06 m)



(b) Aggressive driving behavior: Vehicle 1443 (average gap distance 7.99 m)



(c) Timid driving behavior: Vehicle 3072 (average gap distance 24.87 m)

Fig. 12. Models' performance in terms of driving behaviors.

$$a_n(t+1), \dots, a_n(t+L) = \text{Fun}(\Delta x_n^{n-\theta}(t-T+1), \dots, \Delta x_n^{n-\theta}(t), \\ \Delta v_n^{n-\theta}(t-T+1), \dots, \Delta v_n^{n-\theta}(t), \\ v_n(t-T+1), \dots, v_n(t)) \quad (25)$$

Where $\Delta x_n^{n-\theta}(t)$ denotes the gap distance between the subject vehicle n and the θ ($\theta = 1, 2, \dots, n_a$) nearest preceding vehicle $n-\theta$, and $\Delta v_n^{n-\theta}(t)$ represents the relative speed in the same way.

The platoons simulated by the original and extended seq2seq car-following models are shown in Fig. 15. For the platoon following Vehicle 898, the models considering spatial anticipation perform much better than the original model, as shown in Fig. 15(a) ($MSE = 41.51, 18.12$, and 24.98). For the platoon following Vehicle 2774, the simulation results are comparable with each other, and the extended models turn out to be slightly better than the original model, as shown in Fig. 15(b) ($MSE = 90.78, 86.48$, and 88.72). Besides, in both platoons, the oscillations are not that exaggerated for the simulations with spatial anticipation. Therefore, the seq2seq

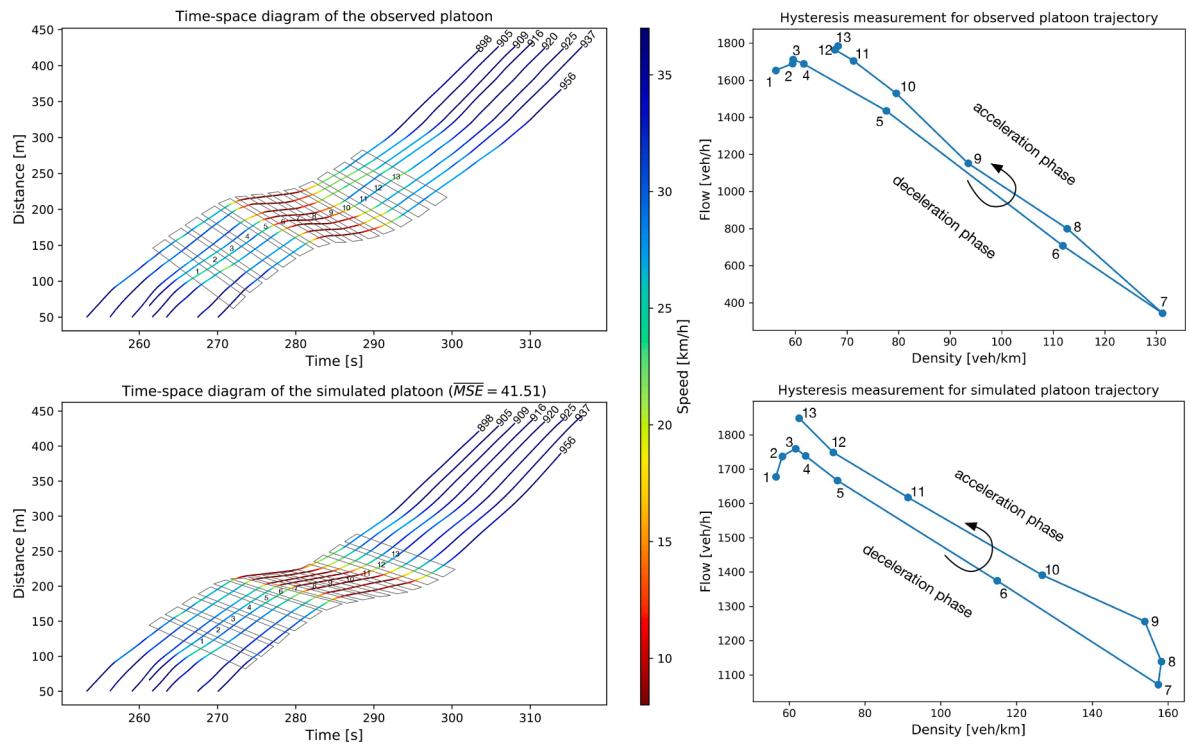


Fig. 13. Comparison of the observed and simulated platoons following Vehicle 898 on Lane 2.

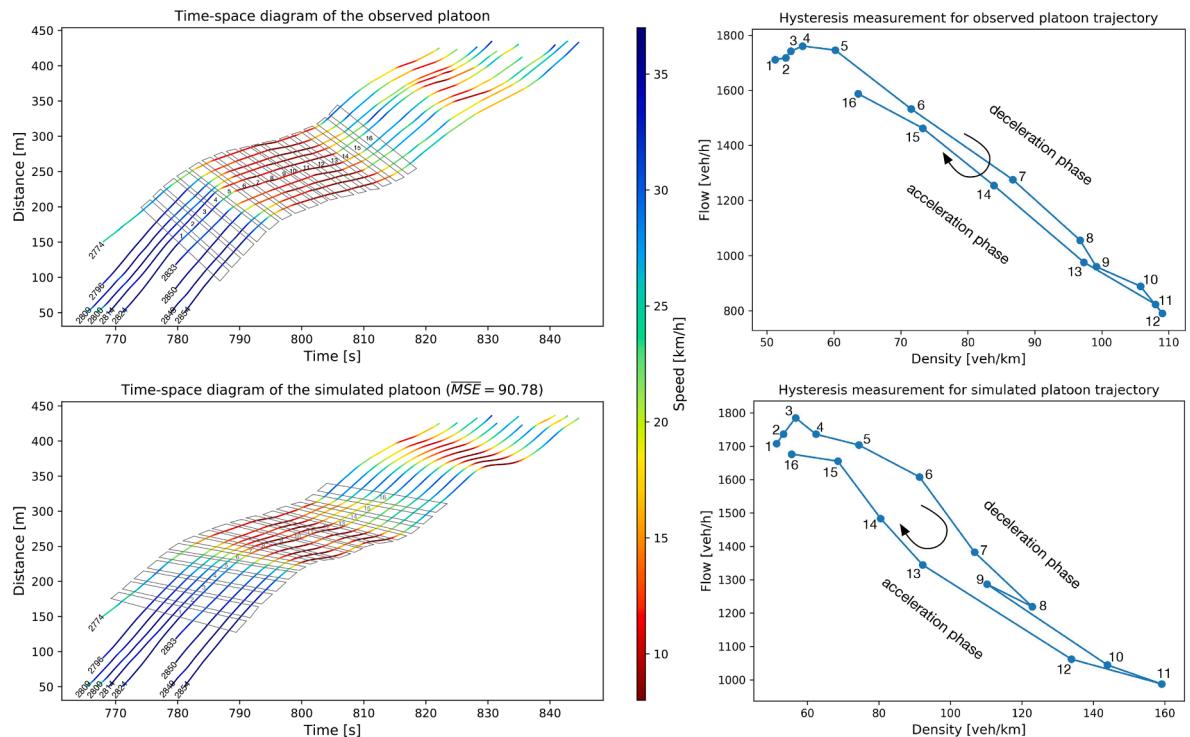
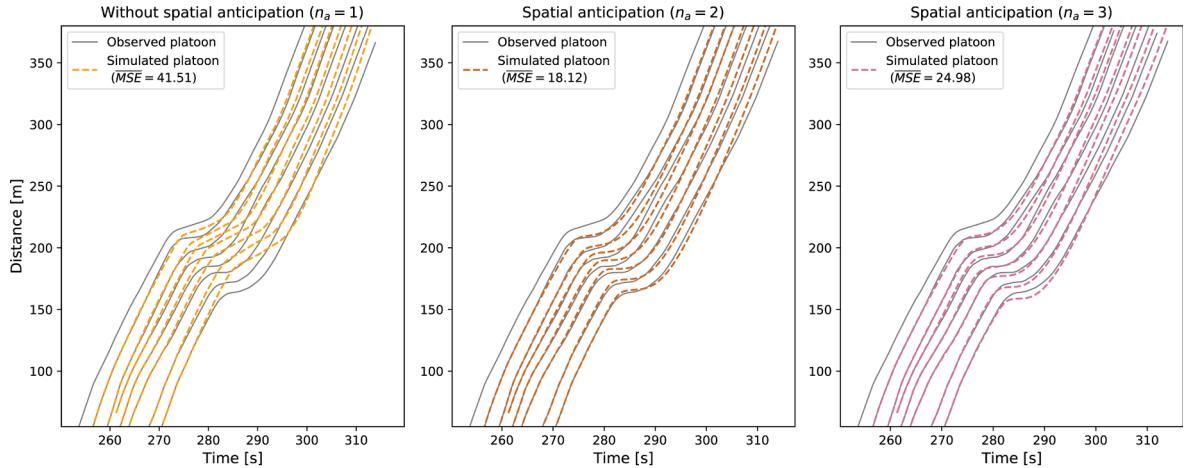
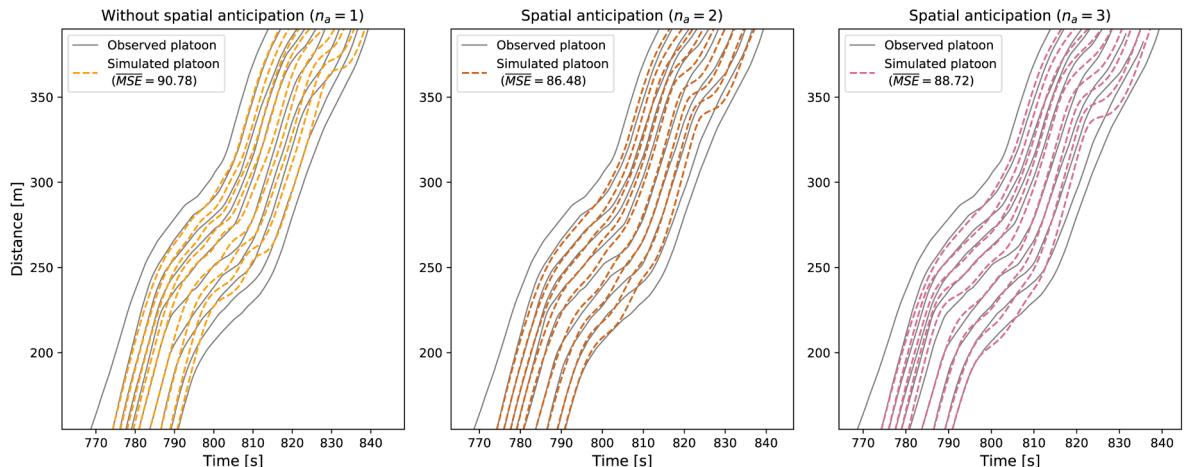


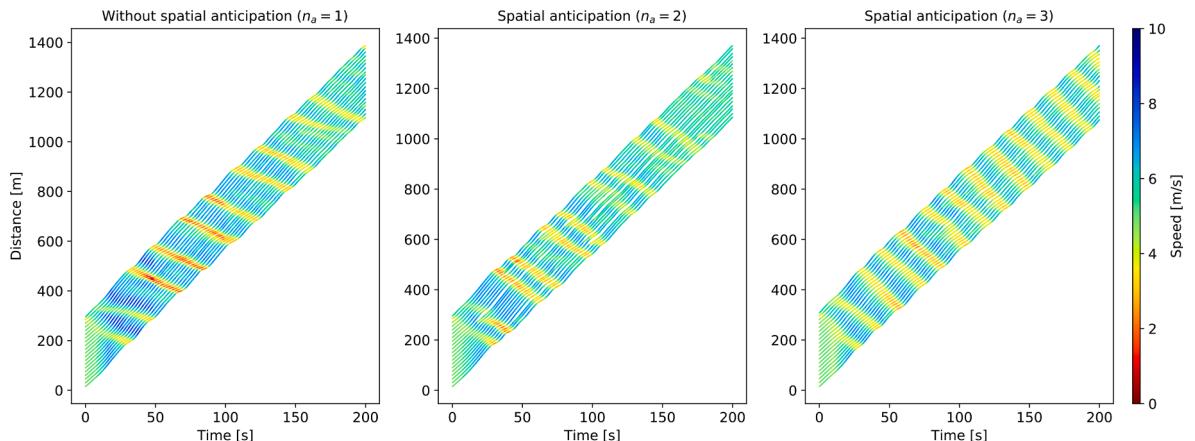
Fig. 14. Comparison of the observed and simulated platoons following Vehicle 2774 on Lane 2.



(a) Platoon following Vehicle 898 on Lane 2



(b) Platoon following Vehicle 2774 on Lane 2

Fig. 15. Platoons simulated by the original and extended seq2seq car-following models.**Fig. 16.** Time–space diagrams of numerical simulations under a periodic boundary condition.

car-following model with spatial anticipation ($n_a > 1$) can improve the performance of platoon simulation, and the model taking the second nearest preceding vehicle into consideration ($n_a = 2$) achieves the best consistency with real trajectories as well as a moderate oscillation.

To further test the stability of extended models, we implement numerical simulations under a periodic boundary condition. We assume there is a 300 m single ring road, and 20 vehicles with lengths of 5 m run on it with an initial speed ($v_0 = 4.6647 \text{ m/s}$). For Vehicle 1, there is an initial disturbance with 5 m backward, then the gap distances of Vehicle 1 and 2 are 15 m and 5 m, respectively. Except for Vehicle 1 and 2, the initial gap distances for other vehicles (from Vehicle 3 to Vehicle 20) are 10 m. Numerical simulations are conducted with original and extended seq2seq car-following models, as the time-space diagrams shown in Fig. 16. We find that the speed fluctuation ranges of extended models ($n_a > 1$) are smaller than that of the original model ($n_a = 1$), which is benefit from the spatial anticipation. Furthermore, the original model ($n_a = 1$) and the extended model ($n_a = 2$) have better performance on oscillation smoothness than the other extended model ($n_a = 3$). Although the time-space diagram of the extended model considering two nearest preceding vehicles ($n_a = 2$) shows asymmetrical gap distances inside the platoon, its speed fluctuation and traffic oscillation are flattened with time to some extent. Therefore, from Fig. 15 and 16, it indicates that incorporating spatial anticipation ($n_a = 2$) into the seq2seq car-following model can improve platoon simulation accuracy and traffic flow stability.

5. Conclusions

In summary, we focus on the study of deep learning car-following models that belong to the data-driven car-following models. The high-fidelity NGSIM data used in this paper are suitable for the training, validation and simulation testing of data-driven models. We investigate the existing neural networks based car-following models and identify the possibility of multi-step predictions for car-following behavior modeling. Thus, the seq2seq car-following model is proposed with the consideration of memory effect and reaction delay. The proposed seq2seq model is compared with the IDM and the LSTM model on trajectory simulation, which is further applied to platoon simulation to reproduce traffic oscillation and hysteresis. The advantages of the proposed seq2seq model can be concluded as follows:

- The model can make multi-step predictions with the consideration of reaction delay.
- The model can better handle the property of different lengths for input and output sequences, as the lengths defined by historical time steps and prediction horizons respectively.
- The trajectories simulated by the seq2seq model are close to the real trajectories, and the model has a stable predictive quality according to the statistical analysis of models' comparison.
- The model can capture heterogeneous driving behaviors by learning the underlying information from the field data.
- The model can well reproduce different levels of hysteresis phenomenon with platoon simulation.
- The model extended with spatial anticipation ($n_a = 2$) can improve platoon simulation accuracy and traffic flow stability.

Based on the estimation and analysis of reaction delay, the proposed seq2seq model incorporates a fixed reaction delay to reflect the decision-making process. However, reaction times in real traffic vary for complex reasons, and the fixed reaction delay could be a limitation. Because of the fixed learning property of artificial intelligence, the deep learning model with instantaneous reaction delay considered needs to be elaborately investigated to keep high prediction accuracy. In addition, with the rapid development of deep learning algorithms, some mechanisms might be able to further promote the model's performance. These are future work directions.

CRediT authorship contribution statement

Lijing Ma: Methodology, Software, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Visualization, Funding acquisition. **Shiru Qu:** Conceptualization, Methodology, Validation, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments. This research has been supported by a key project from the Education Department of Sichuan Province, China (Grant No. 18ZA0302).

References

- Ahmed, K.I., 1999. Modeling drivers' acceleration and lane changing behavior. Ph.D. thesis. Massachusetts Institute of Technology.
- Ahn, S., Vadlamani, S., Laval, J., 2013. A method to account for non-steady state conditions in measuring traffic hysteresis. *Transport. Res. Part C: Emerg. Technol.* 34, 138–147.
- Aycin, M., Benekohal, R., 2004. Performance of the generalized car-following model obtained from macroscopic flow relationships in simulating field data. In: TRB Annual Meeting Washington, DC.
- Bando, M., Hasebe, K., Nakayama, A., Shibata, A., Sugiyama, Y., 1995. Dynamical model of traffic congestion and numerical simulation. *Phys. Rev. E* 51, 1035.
- Bengio, Y., Simard, P., Frasconi, P., et al., 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks* 5, 157–166.
- Bilban, M., Vojvoda, A., Jerman, J., 2009. Age affects drivers' response times. *Collegium Antropol.* 33, 467–471.
- Brackstone, M., McDonald, M., 1999. Car-following: a historical review. *Transport. Res. Part F: Traffic Psychol. Behav.* 2, 181–196.
- Chandler, R.E., Herman, R., Montroll, E.W., 1958. Traffic dynamics: studies in car following. *Oper. Res.* 6, 165–184.

- Chen, D., Ahn, S., Laval, J., Zheng, Z., 2014. On the periodicity of traffic oscillations and capacity drop: the role of driver characteristics. *Transport. Res. Part B: Methodol.* 59, 117–136.
- Chen, D., Laval, J., Zheng, Z., Ahn, S., 2012a. A behavioral car-following model that captures traffic oscillations. *Transport. Res. Part B: Methodol.* 46, 744–761.
- Chen, D., Laval, J.A., Ahn, S., Zheng, Z., 2012b. Microscopic traffic hysteresis in traffic oscillations: A behavioral perspective. *Transport. Res. Part B: Methodol.* 46, 1440–1453.
- Cheng, Q., Liu, Y., Wei, W., Liu, Z., 2016. Analysis and forecasting of the day-to-day travel demand variations for large-scale transportation networks: a deep learning approach. *Transportation Analytics Contest, Tech. Rep.*
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
- Chong, L., Abbas, M.M., Medina, A., 2011. Simulation of driver behavior with agent-based back-propagation neural network. *Transp. Res. Rec.* 2249, 44–51.
- Davis, L., 2003. Modifications of the optimal velocity traffic model to include delay due to driver reaction time. *Physica A* 319, 557–567.
- Edie, L.C., 1963. Discussion of traffic stream measurements and definitions. Port of New York Authority.
- Fambro, D.B., Koppa, R.J., Picha, D.L., Fitzpatrick, K., 1998. Driver perception-brake response in stopping sight distance situations. *Transp. Res. Rec.* 1628, 1–7.
- Fellendorf, M., Vortsch, P., 2010. Microscopic traffic flow simulator vissim. In: *Fundamentals of traffic simulation*. Springer, pp. 63–93.
- FHWA, 2008. The Next Generation Simulation (NGSIM) [Online]. Available: <http://www.ngsim.fhwa.dot.gov/>.
- Fouladgar, M., Parchami, M., Elmarsi, R., Ghaderi, A., 2017. Scalable deep traffic flow neural networks for urban traffic congestion prediction. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 2251–2258.
- Gazis, D.C., Herman, R., Rothery, R.W., 1961. Nonlinear follow-the-leader models of traffic flow. *Oper. Res.* 9, 545–567.
- Gipps, P.G., 1981. A behavioural car-following model for computer simulation. *Transport. Res. Part B: Methodol.* 15, 105–111.
- Gong, H., Liu, H., Wang, B.H., 2008. An asymmetric full velocity difference car-following model. *Physica A* 387, 2595–2602.
- Green, M., 2000. how long does it take to stop? methodological analysis of driver perception-brake times. *Transport. Human Factors* 2, 195–216.
- Gurusinghe, G.S., Nakatsuiji, T., Azuta, Y., Ranjitkar, P., Tanaboriboon, Y., 2002. Multiple car-following data with real-time kinematic global positioning system. *Transp. Res.* 1802, 166–180.
- He, Z., Zheng, L., Guan, W., 2015. A simple nonparametric car-following model driven by field data. *Transport. Res. Part B: Methodol.* 80, 185–201.
- Helbing, D., Tilch, B., 1998. Generalized force model of traffic dynamics. *Phys. Rev. E* 58, 133.
- Helly, W., 1959. Simulation of bottlenecks in single-lane traffic flow.
- Herman, R., 1959. Car-following and steady state flow. *Theory Traffic Flow Symp. Proc.* 1–13.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9, 1735–1780.
- Hoogendoorn, S.P., Ossen, S., Schreuder, M., 2006. Empirics of multianticipative car-following behavior. *Transp. Res. Rec.* 1965, 112–120.
- Huang, X., Sun, J., Sun, J., 2018. A car-following model considering asymmetric driving behavior based on long short-term memory neural networks. *Transport. Res. Part C: Emerg. Technol.* 95, 346–362.
- Jia, H., Juan, Z., Ni, A., 2003. Develop a car-following model using data collected by five-wheel system, in: *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*. IEEE, pp. 346–351.
- Jiang, R., Wu, Q., Zhu, Z., 2001. Full velocity difference model for a car-following theory. *Phys. Rev. E* 64, 017101.
- Johansson, G., Rumar, K., 1971. Drivers' brake reaction times. *Human Factors* 13, 23–27.
- Kehtarnavaz, N., Groswald, N., Miller, K., Lascue, P., 1998. A transportable neural-network approach to autonomous vehicle following. *IEEE Trans. Veh. Technol.* 47, 694–702.
- Khodayari, A., Ghaffari, A., Kazemi, R., Braunstingl, R., 2012. A modified car-following model based on a neural network model of the human driver effects. *IEEE Trans. Syst., Man, Cybernet.-Part A: Syst. Humans* 42, 1440–1449.
- Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- Koutsopoulos, H.N., Farah, H., 2012. Latent class model for car following behavior. *Transport. Res. Part B: Methodol.* 46, 563–578.
- Laval, J.A., 2011. Hysteresis in traffic flow revisited: An improved measurement method. *Transport. Res. Part B: Methodol.* 45, 385–391.
- Laval, J.A., Leclercq, L., 2010. A mechanism to describe the formation and propagation of stop-and-go waves in congested freeway traffic. *Philos. Trans. Royal Soc. A: Math., Phys. Eng. Sci.* 368, 4519–4541.
- Laval, J.A., Toth, C.S., Zhou, Y., 2014. A parsimonious model for the formation of oscillations in car-following models. *Transport. Res. Part B: Methodol.* 70, 228–238.
- LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521, 436.
- Lee, G., 1966. A generalization of linear car-following theory. *Oper. Res.* 14, 595–606.
- Lenz, H., Wagner, C., Sollacher, R., 1999. Multi-anticipative car-following model. *Eur. Phys. J. B-Condensed Matter Complex Syst.* 7, 331–335.
- Li, L., Chen, X.M., Zhang, L., 2016. A global optimization algorithm for trajectory data based car-following model calibration. *Transport. Res. Part C: Emerg. Technol.* 68, 311–332.
- Lin, Y., Wang, P., Zhou, Y., Ding, F., Wang, C., Tan, H., 2019. Platoon trajectories generation: A unidirectional interconnected lstm-based car following model. arXiv preprint arXiv:1910.11843.
- Liu, F., Cheng, R., Ge, H., Yu, C., 2016. A new car-following model with consideration of the velocity difference between the current speed and the historical speed of the leading car. *Physica A* 464, 267–277.
- Ma, X., Andréasson, I., 2006. Driver reaction delay estimation from real data and its application in gm-type model evaluation. *Transp. Res. Rec.* 130–141.
- Magister, T., Krulec, R., Batista, M., Bogdanović, L., 2005. The driver reaction time measurement experiences. *Innovat. Autom. Technol.-IAT* 5.
- Montanino, M., Punzo, V., 2013. Reconstructed NGSIM I80-1. COST ACTION TU0903 - MULTITUDE. Available: <http://www.multitude-project.eu/exchange/101.html>.
- Montanino, M., Punzo, V., 2015. Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. *Transport. Res. Part B: Methodol.* 80, 82–106.
- Morton, J., Wheeler, T.A., Kochenderfer, M.J., 2017. Analysis of recurrent neural networks for probabilistic modeling of driver behavior. *IEEE Trans. Intell. Transp. Syst.* 18, 1289–1298.
- Newell, G.F., 2002. A simplified car-following theory: a lower order model. *Transport. Res. Part B: Methodol.* 36, 195–205.
- Nguyen, H., Kieu, L.M., Wen, T., Cai, C., 2018. Deep learning methods in transportation domain: a review. *IET Intel. Transport Syst.* 12, 998–1004.
- Ozaki, H., 1993. Reaction and anticipation in the car-following behavior. In: *Proc. of 12th International Symposium on Theory of Traffic Flow and Transportation*, pp. 349–366.
- Panwai, S., Dia, H., 2007. Neural agent car-following models. *IEEE Trans. Intell. Transp. Syst.* 8, 60–70.
- Papathanasiou, V., Antoniou, C., 2015. Towards data-driven car-following models. *Transport. Res. Part C: Emerg. Technol.* 55, 496–509.
- Peng, G., He, H., Lu, W.Z., 2016. A new car-following model with the consideration of incorporating timid and aggressive driving behaviors. *Physica A* 442, 197–202.
- Peng, G., Sun, D., 2010. A dynamical model of car-following with the consideration of the multiple information of preceding cars. *Phys. Lett. A* 374, 1694–1698.
- Pipes, L.A., 1953. An operational analysis of traffic dynamics. *J. Appl. Phys.* 24, 274–281.
- Punzo, V., Borzacchiello, M.T., Ciuffo, B., 2011. On the assessment of vehicle trajectory data accuracy and application to the next generation simulation (ngsim) program data. *Transport. Res. Part C: Emerg. Technol.* 19, 1243–1262.
- Ranjitkar, P., Nakatsuiji, T., Azuta, Y., Gurusinghe, G.S., 2003. Stability analysis based on instantaneous driving behavior using car-following data. *Transp. Res.* 1852, 140–151.
- Saifuzzaman, M., Zheng, Z., 2014. Incorporating human-factors in car-following models: a review of recent developments and research needs. *Transport. Res. Part C: Emerg. Technol.* 48, 379–403.
- Sharma, A., Zheng, Z., Kim, J., Bhaskar, A., Haque, M.M., 2019. Estimating and comparing response times in traditional and connected environments. *Transp. Res.* Rec. 0361198119837964.

- Sipahi, R., Niculescu, S.I., 2010. Stability of car following with human memory effects and automatic headway compensation. *Philos. Trans. Royal Soc. A: Math., Phys. Eng. Sci.* 368, 4563–4583.
- Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. *Adv. Neural Inform. Process. Syst.* 3104–3112.
- Tang, T., Huang, H., Zhao, S., Xu, G., 2009. An extended ov model with consideration of driver's memory. *Int. J. Mod. Phys. B* 23, 743–752.
- Toledo, T., 2007. Driving behaviour: models and challenges. *Transp. Rev.* 27, 65–84.
- Toledo, T., Koutsopoulos, H.N., Ahmed, K.I., 2007. Estimation of vehicle trajectories with locally weighted regression. *Transp. Res. Rec.* 1999, 161–169.
- Treiber, M., Helbing, D., 2003. Memory effects in microscopic traffic models and wide scattering in flow-density data. *Phys. Rev. E* 68, 046119.
- Treiber, M., Hennecke, A., Helbing, D., 2000. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E* 62, 1805.
- Treiber, M., Kesting, A., 2013. Microscopic calibration and validation of car-following models—a systematic approach. *Proc.-Soc. Behav. Sci.* 80, 922–939.
- Treiber, M., Kesting, A., Helbing, D., 2006. Delays, inaccuracies and anticipation in microscopic traffic models. *Physica A* 360, 71–88.
- Treiterer, J., Myers, J., 1974. The hysteresis phenomenon in traffic flow. *Transport. Traffic Theory* 6, 13–38.
- Triggs, T.J., Harris, W.G., 1982. Reaction time of drivers to road stimuli.
- Wang, X., Jiang, R., Li, L., Lin, Y., Zheng, X., Wang, F.Y., 2018. Capturing car-following behaviors by deep learning. *IEEE Trans. Intell. Transp. Syst.* 19, 910–920.
- Wang, X., Jiang, R., Li, L., Lin, Y.L., Wang, F.Y., 2019. Long memory is important: A test study on deep-learning based car-following model. *Physica A* 514, 786–795.
- Wei, D., Liu, H., 2013. Analysis of asymmetric driving behavior using a self-learning approach. *Transport. Res. Part B: Methodol.* 47, 1–14.
- Wiedemann, R., 1974. Simulation of road traffic flow. In: Reports of the Institute for Transport and Communication. University of Karlsruhe.
- Yeo, H., 2008. Asymmetric microscopic driving behavior theory.
- Yu, S., Shi, Z., 2015. Dynamics of connected cruise control systems considering velocity changes with memory feedback. *Measurement* 64, 34–48.
- Zhang, X., 2014. Empirical analysis of a generalized linear multianticipative car-following model in congested traffic conditions. *J. Transport. Eng.* 140, 04014018.
- Zhang, X., Bham, G.H., 2007. Estimation of driver reaction time from detailed vehicle trajectory data. *Moas* 7, 574–579.
- Zhao, Z., Chen, W., Wu, X., Chen, P.C., Liu, J., 2017. Lstm network: a deep learning approach for short-term traffic forecast. *IET Intel. Transport Syst.* 11, 68–75.
- Zheng, J., Suzuki, K., Fujita, M., 2013. Car-following behavior with instantaneous driver-vehicle reaction delay: A neural-network-based methodology. *Transport. Res. Part C: Emerg. Technol.* 36, 339–351.
- Zheng, Z., Ahn, S., Chen, D., Laval, J., 2011. Freeway traffic oscillations: microscopic analysis of formations and propagations using wavelet transform. *Proc.-Soc. Behav. Sci.* 17, 702–716.
- Zhou, M., Qu, X., Jin, S., 2016. On the impact of cooperative autonomous vehicles in improving freeway merging: a modified intelligent driver model-based approach. *IEEE Trans. Intell. Transp. Syst.* 18, 1422–1428.
- Zhou, M., Qu, X., Li, X., 2017. A recurrent neural network based microscopic car following model to predict traffic oscillation. *Transport. Res. Part C: Emerg. Technol.* 84, 245–264.
- Zhu, M., Wang, X., Hu, J., 2020. Impact on car following behavior of a forward collision warning system with headway monitoring. *Transport. Res. Part C: Emerg. Technol.* 111, 226–244.
- Zhu, M., Wang, X., Wang, Y., 2018. Human-like autonomous car-following model with deep reinforcement learning. *Transport. Res. Part C: Emerg. Technol.* 97, 348–368.