Transportation Research Part C 104 (2019) 287–304

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Simultaneous modeling of car-following and lane-changing behaviors using deep learning

Xiaohui Zhang, Jie Sun, Xiao Qi, Jian Sun*

Department of Traffic Engineering & Key Laboratory of Road and Traffic Engineering, Ministry of Education, Tongji University, Shanghai, China

ARTICLE INFO

Keywords:
Traffic flow
Car-following
Lane-changing
Simultaneous modeling
Deep learning
Long Short-Term Memory

ABSTRACT

Car-following (CF) and lane-changing (LC) behaviors are two basic movements in traffic flow which are generally modeled separately in the literature, and thus the interaction between the two behaviors may be easily ignored in separated models and lead to unrealistic traffic flow description. In this paper, we adopt a deep learning model, long short-term memory (LSTM) neural networks, to model the two basic behaviors simultaneously. By only observing the position information of the six vehicles surrounding the subject vehicle, the LSTM can extract the significant features that influence the CF and LC behaviors automatically and predict the vehicles behaviors with time-series data and memory effects. In addition, we propose a hybrid retraining constrained (HRC) training method to further optimize the LSTM model. With the I-80 trajectory data of NGSIM dataset we train and test the HRC LSTM model, while the results show that HRC LSTM model can accurately estimate CF and LC behaviors simultaneously with low longitudinal trajectories error and high LC prediction accuracy compared with the classical models. We also evaluate the transferability of the proposed model with the US101 dataset and a good transferability result is obtained as well.

1. Introduction

As two fundamental driving behaviors in traffic flow, Car-Following (CF) and Lane-Changing (LC) maneuvers which describe vehicular longitudinal and lateral interactions on the road respectively, have attracted long term interests in the traffic flow research community. For CF behavior, it has been widely studied for many years and numerous CF models have been developed to depict CF behaviors in terms of different perspectives such as engineering perspective and human perspective. For a detailed review, please refer to Brackstone and McDonald (1999), Toledo (2007) and Saifuzzaman and Zheng (2014). With respect to LC behavior, the modeling efforts are not as many as CF due to the complexity of LC behavior. More specifically, LC is often categorized into different types (e.g., mandatory LC and discretionary LC; or free, cooperative, and forced LC (Hidas, 2002; Sun et al., 2014a)). In addition, the whole LC process can be divided into four sub-models, including motivation generation of LC, lane choice model, gap acceptance model, and LC execution model (Toledo et al., 2007), while most previous studies only focus on the gap acceptance model (Ahmed, 1999). However, because of its negative influence on traffic flow, incorporating LC maneuvers in traffic flow theory and modeling is of vital importance (Hidas, 2005; Sun et al., 2014a, 2014b). Recent developments of LC modeling is reviewed in detail by Zheng (2014).

Generally, CF and LC behaviors are studied and modeled separately in the literature for the sake of simplicity and focus. However, from the human perspective, drivers can hardly split the driving task into two independent tasks during driving process, since CF and

* Corresponding author.
E-mail address: sunjian@tongji.edu.cn (J. Sun).

https://doi.org/10.1016/j.trc.2019.05.021
Received 5 June 2018; Received in revised form 6 April 2019; Accepted 17 May 2019
Available online 22 May 2019
0968-090X/ © 2019 Elsevier Ltd. All rights reserved.

LC tasks have a significant impact on each other (Toledo, 2007). Although it is intuitive to consider CF and LC as an indivisible process, integrating them into one model is still challenging for several reasons: (1) The influencing factors for the two behaviors are numerous but different. For example, the gap between the subject vehicle and preceding vehicle in the current lane and the speeds of subject vehicle and preceding vehicle are commonly-adopted variables in CF modeling, whereas LC modeling usually considers the gap and speeds of vehicles in the current and target lanes at the same time. (2) Driver's decisions on CF and LC are always dependent on the historical driving behaviors and the past traffic states (Toledo, 2007), while mathematical models are not suitable for such consideration. (3) The output of CF model is a continuous variable (e.g. acceleration, or speed) and the result of LC model is a discrete variable (e.g. whether to change lane or not) that are distinct from the former. Thus, only very few studies attempted to develop integrated models for CF and LC behaviors. Toledo et al. (2007) integrated CF model and LC model together by assigning different sub-tasks in different layers. The decision of LC, select of appropriate lane and gap and change of speed are decided step by step according to a large set of sub-models, making the integrated model quite complicated and inflexible. Tomar et al. (2010) tried to predict the LC and CF trajectories with multilayer perceptron neural network (MLP-NN). The results showed that an MLP-NN can give the future path accurately only for discrete patches of the trajectory but not over the complete trajectory. In summary, how to precisely and completely describe the CF and LC behaviors at the same time remains elusive.

However, the CF and LC behaviors are both implied in trajectories. Thus modeling CF and LC behaviors can be equivalent to trajectory prediction, which should meet requirement in terms of microscopic level (e.g. CF and LC behaviors). In the domain of self-driving vehicles, there are four steps including perceiving the environment, processing that information, predicting how others in that environment will behave and finally making driving decisions (Waymo, 2018). Therefore, many studies attempted to focus on two-dimensional trajectory prediction to mimic human driving behavior (Cosgun et al., 2017; Djuric et al., 2018; Li et al., 2017; Woo et al., 2017; Zhang et al., 2017)

On the other hand, machine learning methods have been widely adopted for CF and LC modeling because of the easy access to massive high-resolution trajectories data and recent development of data mining technologies. A number of data-driven models using shallow learning were developed for CF models (Khodayari et al., 2012; Panwai and Dia, 2007; Wang et al., 2018; Wei and Liu, 2013; Zheng et al., 2013; Zhou et al., 2017) and LC models (Hunt and Lyons, 1994; Kita, 1999; Kita et al., 2002; Moridpour et al., 2009; Talebpour et al., 2015; Wu et al., 2000), respectively. Previous studies also demonstrated better performance of data-driven models than conventional analytical models in multi-aspects, such as the accuracy of trajectories and the replication of traffic flow characteristics. Among these studies, the Neutral Network (NN) method with different variations is commonly used. In general, human driving behaviors are highly non-linear and complex maneuvers, which is difficult to be modeled by mathematical models or conventional shallow machine learning models. The recent emergence of deep learning technology has been shown to be a highly effective learning approach and has demonstrated superior performance in various domains (e.g., traffic speed prediction, driving distraction classification and CF behavior modeling) (Huang et al., 2018; Ma et al., 2015; Wang et al., 2018; Wollmer et al., 2011). Therefore, in this paper, we attempt to reproduce the vehicles' complete trajectories by including the CF and LC behaviors in one deep learning model.

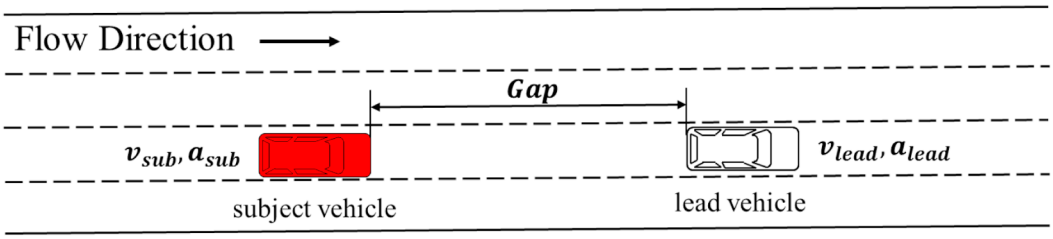
Considering that Long Short-Term Memory neural networks (Hochreiter and Urgen Schmidhuber, 1997), a special Recurrent Neural Networks, is able to learn the time series with long time spans and automatically determine the optimal time lags for prediction, we adopt the LSTM to develop the integrated CF and LC model by learning human decision-making experience in driving. By only observing the position information of the six vehicles surrounding the subject vehicle, the LSTM can extract the significant features that influence the CF and LC behaviors automatically and predict the vehicles behaviors with time-series data and memory effects. In addition, to promote the performance of LSTM model, we further propose a novel hybrid retraining constrained (HRC) training method. With the I-80 dataset in NGSIM dataset, we first train and test the HRC LSTM model and compare it with the field data. And we also assess the transferability of the model with the US-101 dataset to examine its ability to implement directly on other sites.

The remaining part of this paper is organized as follows: Section 2 proposes the HRC training method based on the LSTM model; Section 3 evaluates the HRC LSTM model with I-80 dataset; Section 4 studies the transferability of proposed model and Section 5 summarizes main conclusions and discusses the future study.

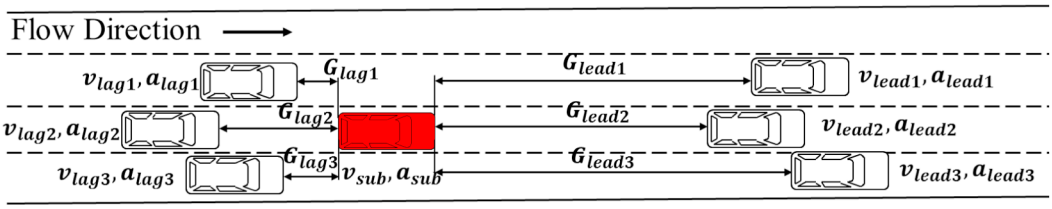
2. Methodology

As mentioned before, three challenges exist when we integrate the CF and LC modeling into one model. The most critical one is the different influencing factors for the two behaviors. More specifically, CF and LC are two completely different driving behaviors and many different factors influence the two behaviors. As shown in Fig. 1, CF models usually consider 6 parameters, such as speed (v_{sub}, v_{lead}), acceleration (a_{sub}, a_{lead}) and gap (time gap, space gap) of subject vehicle and lead vehicle (Saifuzzaman and Zheng, 2014; Treiber et al., 2000). However, LC models usually consider a total of more than 10 parameters (Laval and Daganzo, 2006; Zheng, 2014; Sun et al., 2014a). Compared with CF models, LC models generally take the speeds (v_{sub}, v_{lag1}, v_{lag2}, v_{lag3}, v_{lead1}, v_{lead2}, v_{lead3}) and accelerations (a_{sub}, a_{lag1}, a_{lag2}, a_{lag3}, a_{lead1}, a_{lead2}, a_{lead3}) of vehicles in adjacent lanes, and the gap (G_{lag1}, G_{lag2}, G_{lag3}, G_{lead1}, G_{lead2}, G_{lead3}) between the lead vehicle and lag vehicle in the adjacent lanes into consideration. All above parameters have different effects on modeling CF and LC behaviors and the integrated settings of model input need to be reconsidered when simultaneously modeling. A direct way to incorporate the CF and LC behaviors in one integrated model is to extract the information from the two-dimensional (i.e. longitudinal and lateral) positions of surrounding vehicles and predict the corresponding positions of the subject vehicle taking the advantages of deep learning model. Thus the CF and LC behavior characteristics can be further extracted and studied.

In addition, driving behavior is a time-series process that has fading memory effect (Treiber et al., 2000). When developing an



(a) Main factors considered in CF models



(b) Main factors considered in LC models

Fig. 1. Main factors for CF and LC models.

integrated model including CF and LC behaviors, we should also notice the impact of historical behavior on the current state. Considering the superiority of LSTM on dealing with massive high-dimensional data and fading memory effect, we employ LSTM as the basic model for simultaneous CF and LC modeling. To reduce the cumulative errors in the whole process, we develop a new training method, i.e. HRC LSTM, to further optimize the LSTM model.

The general structure of HRC LSTM model is shown in Fig. 2. The high-dimensional historical time-series position data are used as the input. The complete trajectories are first predicted with the initial LSTM model and then used for retraining. Combined with the safety constraint, the HRC LSTM model is trained with the Train-Predict-Retrain process. With the obtained HRC LSTM model, the positions of subject vehicle in the testing dataset are predicted. The structure of the basic LSTM model is introduced in detail in Section 2.1. The configuration of LSTM neural network is then presented in Section 2.2. The improved HRC training method is explained in Section 2.3.

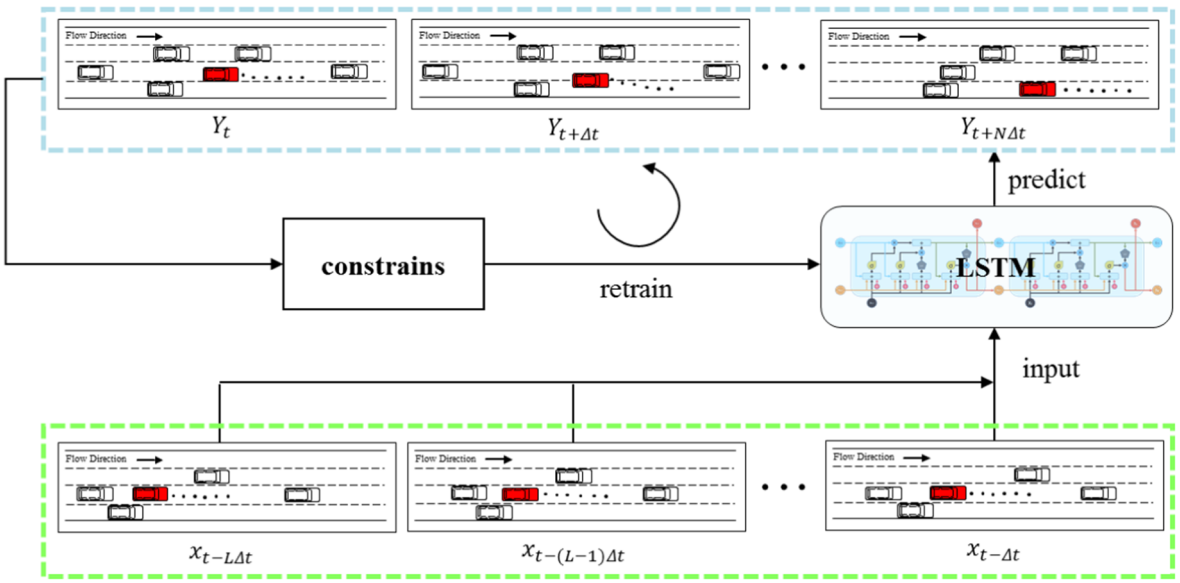


Fig. 2. Model structure of HRC LSTM (L steps refer to the historical length of input and the N steps refer to the output length of a complete trajectory).

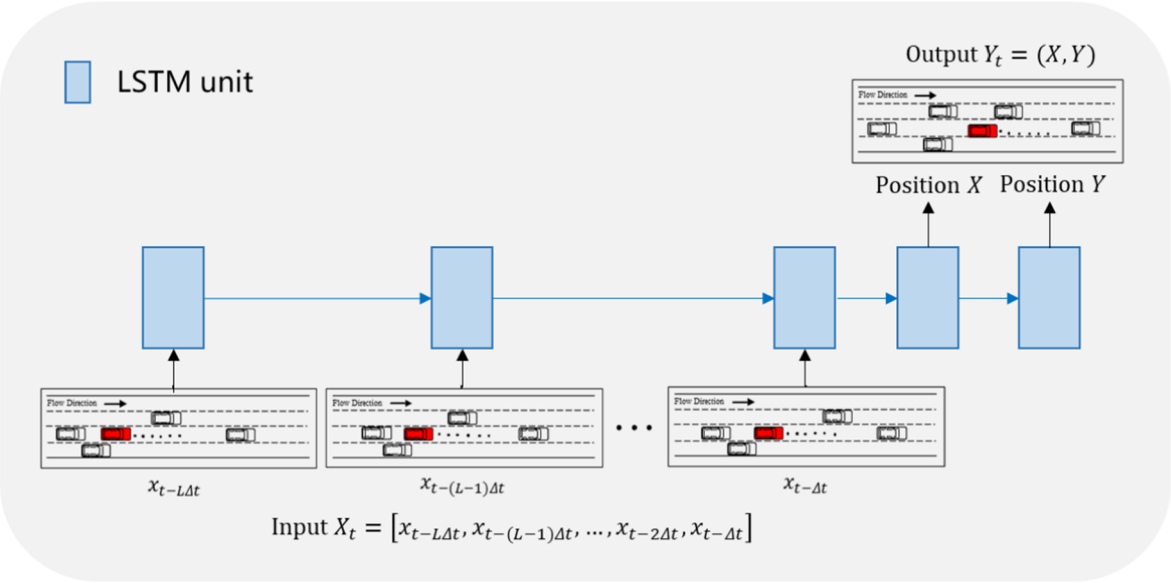


Fig. 3. Model structure of a basic LSTM.

2.1. The basic LSTM model

Recent advances in deep learning have made it possible to extract high-level features from raw data (e.g. images, speech) and LSTM is a special recurrent neural network in essence (LSTM is inherently a deep learning method despite the number of hidden layers, for a signal may propagate through a layer more than once, and the CAP depth is potentially unlimited (Schmidhuber, 2015) in recurrent neural networks like LSTM). So LSTM can obtain the feature of driving situation at different timestamp, and then it can mine higher-level time series features which represent driving memory effects. Due to the advantage of incorporating historical driving information, LSTM model can achieve better prediction.

The structure of a basic LSTM model is shown in Fig. 3. It contains one input layer, one hidden layer and one output layer. The input layer initializes input data for subsequent layers, while the output layer is used for the feature regression. The hidden layer is mainly used for the feature learning of the input data, where the main unit is memory block (see Fig. 4). The input, output and LSTM unit are explained in detail later.

2.1.1. LSTM unit

LSTM with forget gates (Gers et al., 2000) that recognized as the basic LSTM is built first in this work. Every unit of LSTM network

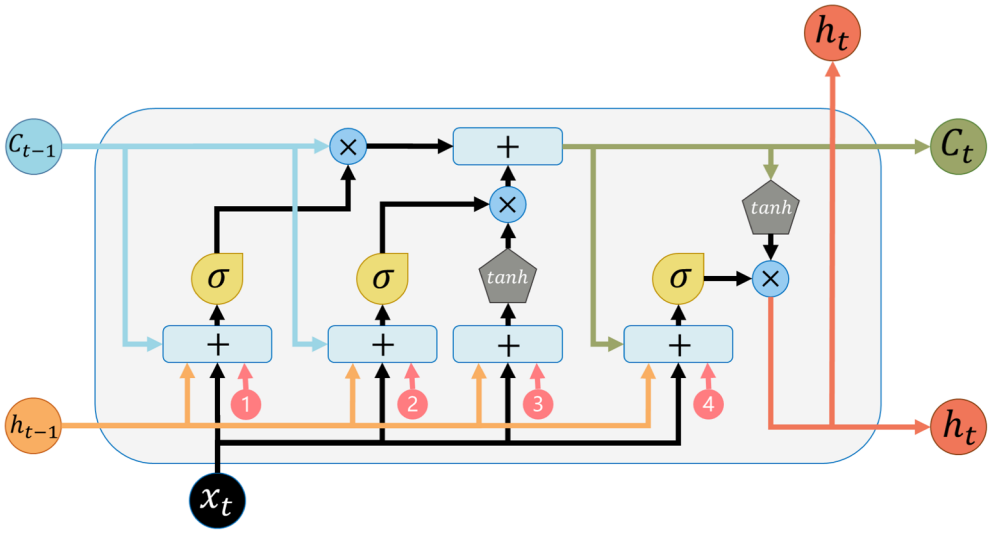


Fig. 4. Schematic diagram of the memory block of LSTM.

is a memory block (as shown in Fig. 4) with three self-parameterized controlling gates, the input, forget and output gate, respectively. In a training cycle, input will remain in the memory of the cell if the input gate is activated. After that, forget gate will decide whether to forget the past value. Finally, the extent to which output in the cell is used to compute the final state is also controlled by the output gate. The compact form of the equations is shown as below, which are also marked out in Fig. 4.

a. **Forget gate**

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{1}$$

b. **Input gate**

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{2}$$

$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{3}$$

c. **Output gate**

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{4}$$

$$h_t = o_t tanh(c_t) \tag{5}$$

where x_t is the input data, σ is the logistic sigmoid function, and i, f, c, o are input gate, forget gate, cell state vectors and output gate, respectively. And they are the same size as the hidden vector h. W_* ($* = xi, hi, ci, xf, hf, cf, xc, hc, xo, ho, co$) stands for the weight matrix from the cell to gate, and b_* ($* = i, f, c, o$) denotes the bias vector of each gate.

2.1.2. Input and output

The input of the integrated model should fully consider the influence of other vehicles in the driving environment and the impact of historical driving behavior. Other than modeling lane-changing (LC)'s decision-making only, we study on two-dimensional trajectory prediction, thus no efforts are needed to determine the initial lane-changing states or distinguish lane-changing types (Li et al., 2018; Wang et al., 2014). For each vehicle, the two-dimensional positions (longitudinal and lateral position) of the closest preceding and following vehicles within detection region in the current lane and adjacent lanes are extracted from NGSIM dataset. The time-sequence historical position information of subject vehicle and six surrounding vehicles are the sole inputs of HRC LSTM model (see Fig. 5), while the two-dimensional positions of subject vehicle in the following time steps are predicted.

Let $x_t \in \mathbb{R}^D$ be the D-dimension vector of the t-th time step and x_t is composed of the position of subject vehicle and positions of detecting vehicles in the detection region. The area which contains 100 m behind the front and a total of three lanes at center of the vehicle are the detection region (as shown in Fig. 5). The nearest vehicle in front of or behind the subject vehicle in each lane can be detected, and the composition of the feature vector x_t in t-th time step is as follows:

$$x_t = [P^t_{sub}, P^t_{lead1}, P^t_{lead2}, P^t_{lead3}, P^t_{lag1}, P^t_{lag2}, P^t_{lag3}] \tag{6}$$

P^t denotes the position of a vehicle in the t-th time step, which includes local x (lateral position) and local y (longitudinal position). It should be noted that there may be no vehicles in the detecting area. If this is the case, the default positions of the vehicle in each lane is set at both ends of the detection area.

Therefore, the input matrix X_t of model training and predicting are composed of feature vectors in L time steps Δt before t-th time step. The expression for X_t is as follows:

$$X_t = [x_{t-L\Delta t}, x_{t-(L-1)\Delta t}, \cdots, x_{t-2\Delta t}, x_{t-\Delta t}] \tag{7}$$

The existing studies on the CF and LC behaviors consider many factors such as speed, acceleration, time gap and space gap et al. Although the feature vector x_t in this study only has information of relative position, the input matrix X_t is composed of feature

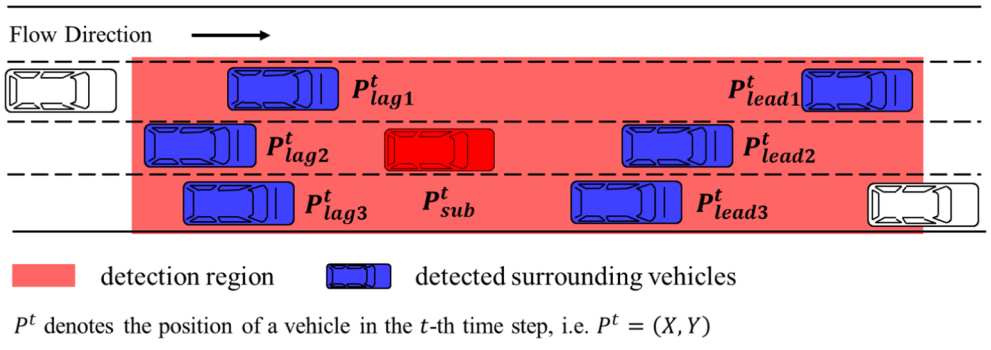


Fig. 5. Model input of HRC LSTM.

vectors at multiple moments. Thus LSTM can be used to characterize speed, acceleration, and other parameters during continuously detecting. So this study only considers the position information of the surrounding vehicles and the subject vehicle in the model input. This kind of inputs ensure the maximum flexibility for the subject vehicle's detecting capabilities if consider how to implement for autonomous vehicles in future.

The model output Y_t corresponding to the input X_t at t-th time step is the predictive position of the subject vehicle at t-th time step, which equals to $\hat{P}_{sub}^{t}$.

In the context of this paper, for every single time step of prediction, x_t is the input data corresponding to the time-sequence historical position information of subject vehicle and six surrounding vehicles, c_t is the internal state vectors and h_t is the hidden vector in the t-th unit. Moreover, as shown in Fig. 3, c_t and h_t are the inputs for the next unit, the output h_t of last two unit can derive the two-dimensional positions of subject vehicle by applying activation function sigmoid.

2.2. Configuration of LSTM neural network

After developing the basic structure of LTSM, several configurations of LSTM need to be specified to optimize the proposed model, such as the loss function, activation function and optimization algorithm.

2.2.1. Loss function

The training process of LSTM is essentially the process of back propagation of the loss function to adjust weights and parameters.

As for the loss function of deep neural network (DNN) model, it quantifies how well the algorithm models the training data, while the training goal is to find optimal parameters to minimize the loss function. Loss functions in machine learning can be broadly divided into classification and regression loss. The general loss functions for classification are log loss, focal loss, KL divergence, exponential loss, hinge loss, etc., while for regression which our study belongs to, MSE (Mean Square Error), MAE (Mean Absolute Error) and huber loss are usually used (Janocha and Czarnecki, 2017). Among them, MSE is the most commonly used regression loss function, which is also adopted as the loss function of HRC-LSTM in this paper. The expression of MSE is as follows:

$$MSE = \frac{\sum_{i=1}^{N} |\hat{P}_{sub}^{i} - P_{sub}^{i}|}{N} \tag{8}$$

where $\hat{P}_{sub}^{i}$ is the i-th sample (the predictive position of subject vehicle, a vector including local x and local y) calculated by model, P_{sub}^{i} is the observed i-th sample, N is the total sample size and $|\hat{P}_{sub}^{i} - P_{sub}^{i}|$ denotes the vector module of the two vectors.

2.2.2. Activation function

The activation function can solve the drift problem of internal states at the beginning of learning, which can improve the ability of nonlinear modeling (Hochreiter and Urgen Schmidhuber, 1997). Sigmoid in Eq. (9) is used as the activation function for the output layer.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{9}$$

2.2.3. Optimization algorithm

The optimization algorithm is a method for gradient descent (Hochreiter and Urgen Schmidhuber, 1997). A method of Adam for stochastic optimization are used. The value of learning rate in Adam are set as 0.001 according to Kingma and Ba (2015). And the batch size is set as 32 after independent experiments.

2.3. The HRC LSTM

In previous studies (Ben Taieb et al., 2012), LSTM is known to be inaccurate for prediction in relatively long-term time steps and can just perform well in the next few time steps. In other words, by inputting all sequences of real data, the next time step prediction is obvious to be accurate. However, when the model is performing continuous prediction, the model obtains complete trajectory prediction results only with the input of initial time period, while the output of each time step of the model will be used as the input for the next time step, thus leading to the error of the model will also gradually accumulate. Therefore, the HRC LSTM is put forward which incorporates retraining and constraints based on the original LSTM model. The whole process of HRC LSTM is shown as Fig. 6. The dataset is divided into D_1 and D_2. Dataset D_1 is used to obtain original LSTM (step ①). Dataset D_2 is used to generate predicted results using original LSTM (step ②). Constraints are added to ensure the safety distance of predicted results, which generates a new dataset D_3 (step ③). Then dataset D_1 and D_3 are merged (step ④) to retrain the LSTM model (step ⑤). The overall error of LSTM model will be reduced by retraining, and the model can be trained iteratively for many epochs until the model error meets the predicted demand (step ⑥). Detailed methods of retraining and constrains are introduced in the following two sections.

2.3.1. Retraining with predictions

Regarding the retraining, it is a kind of disagreement-based learning which could be traced back to the disagreement-based semi-supervised learning. In the paradigm of disagreement-based semi-supervised learning, multiple learners are generated by the labeled

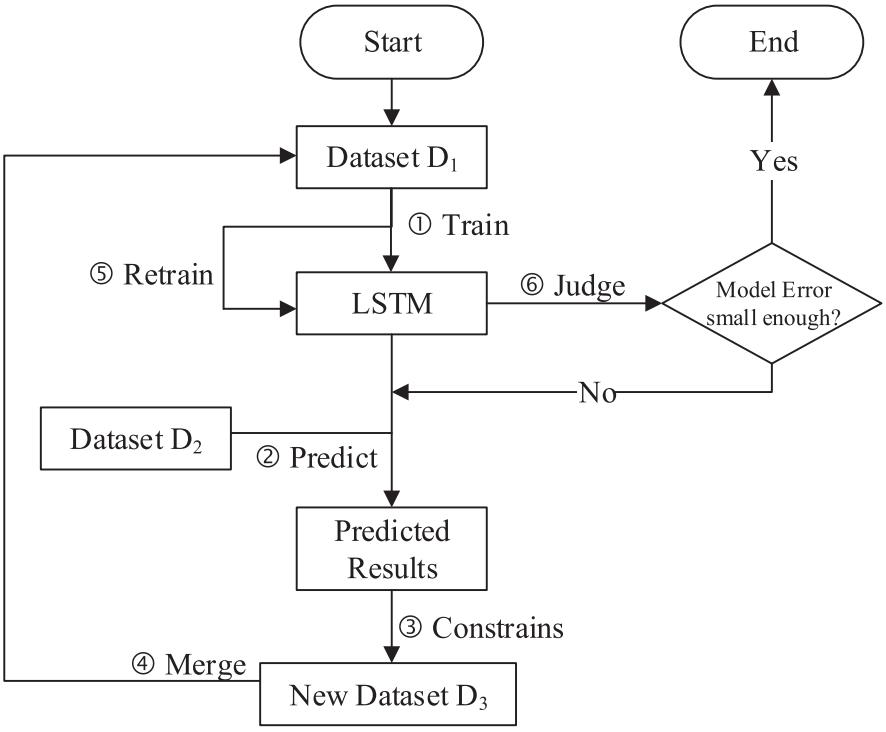


Fig. 6. The process of HRC LSTM.

dataset and collaborate them to exploit unlabeled examples, and then refine themselves through the large disagreement between the multiple learners. In detail, every learner converts the most confidently predicted unlabeled example into a labeled training example to teach other leaners, and finally we can get a fine-tuned model until learners agree on each other (Blum and Mitchell, 1998; Zhou and Li, 2005; Zhou and Li, 2010). Inspired by the disagreement-based semi-supervised learning (learning with single view multiple regressors), retraining utilizes the disagreement between training and retraining to obtain effective generalization performance. In each iteration, a learner predicts new trajectory sequences, while a new learner is trained in next iteration, where the inputs are composed by predicted trajectory sequences in the previous iteration and original trajectories (observed data), and the outputs are the corresponding positions in the observed trajectory data. Note that as every element in a trajectory sequence is both input and output of LSTM, disagreement between training and retraining applies in the input for training a new learner. This process differs from the original disagreement-based semi-supervised learning, where multiple learners predict respective labels for unlabeled examples and disagreement from it consists in the output label of training. Another difference is that there are no unlabeled examples in retraining. Thus the effect of each update can be verified immediately and enhanced in following training by the cycle of Train-Predict-Retrain.

More specifically, as shown in Fig. 7, half of the original training set D_1 is used to obtain an initial LSTM model, and then another dataset D_2 is used as model input to predict by the initial model. Dataset D_2 is partly replaced by the predicted result of D_2 which forms a new dataset D_3. Thus the retraining dataset D^* is formed, where the input of the dataset is composed of half of the original input of training set D_1 and the prediction results from the dataset D_3, and the output remains the same as observed data. After the initial model retrained with the new training set D^*, a new LSTM model is obtained.

This approach has several advantages over conventional training. First, it is readily comprehensible that retraining with predictions will improve model performance, and mixing with half input of real data prevents model from getting more and more biased. Second, the effect of each update can be verified immediately and enhanced in following training by the cycle of Train-Predict-Retrain. In this way, prediction accuracy can be increased to meet the requirement of long-term prediction.

2.3.2. Longitudinal constrains

In the perspective of machine learning, data editing is a useful technique which filters out the training examples wrongly generated in the human labeling process to improve the quality of the training (Li and Zhou, 2005; Wilson, 1972). Thus we proposes the longitudinal constrains in order to ensure the safety distance of vehicles. Particularly, in addition to the vehicle's own dynamics and kinematics constraints, the model should also consider safety constraints of vehicle movement. In practice, vehicle locations are limited by surrounding vehicles, while errors in LSTM prediction may lead to inconsistencies with the actual situation. Therefore, the constraints should be added in the retraining process to ensure the safety of the prediction results. The role of the constraints is shown in Fig. 8.

The retraining process of HRC LSTM is as the same as that in the Section 2.3.1, while the difference is the predictions of

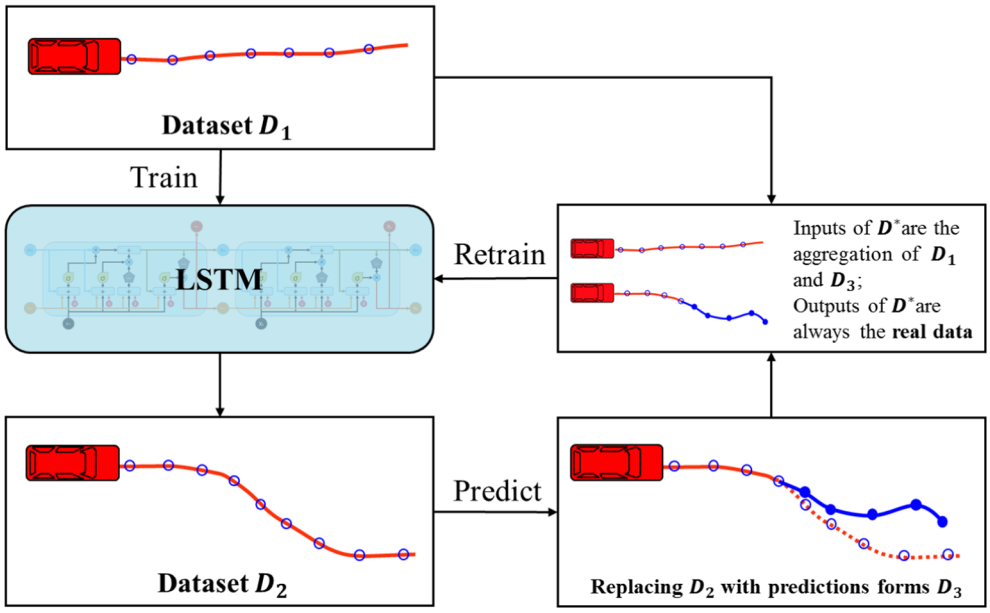


Fig. 7. Schematic diagram of retraining LSTM.

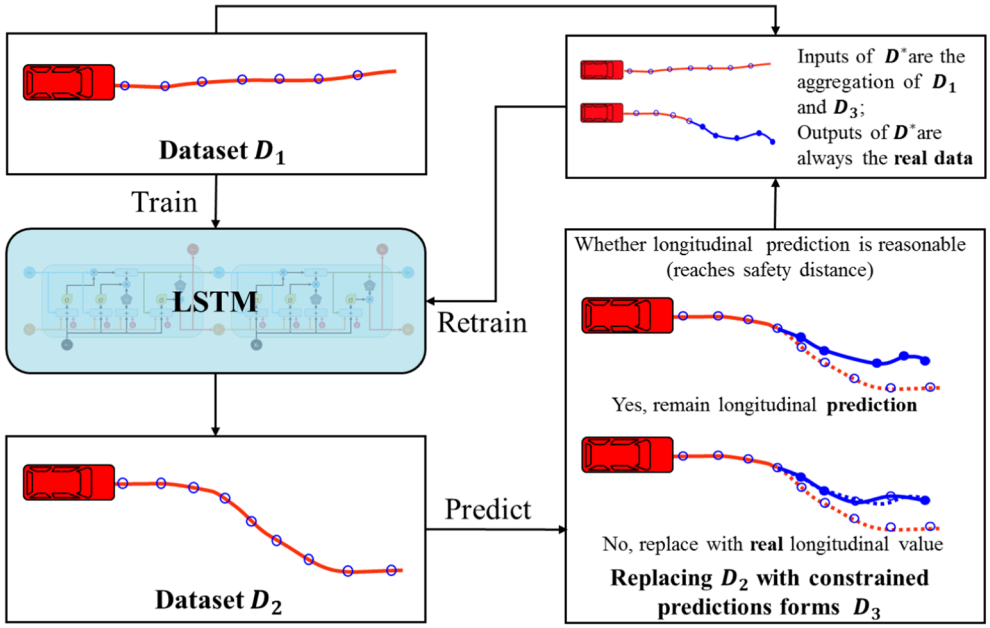


Fig. 8. Schematic diagram of constraints of HRC LSTM.

longitudinal positions should be first assessed to see whether reasonable (meeting the safety condition) or not before retraining. The real longitudinal value of the vehicle will be adopted for the input of retraining if the longitudinal prediction is not reasonable. As per the minimum maintained time headway of human drivers in previous study (Vogel, 2003), we set up the safety constraint that time headway should be larger than 1 s. The constraints are defined in Eq. (10) and Fig. 9.

$$\widehat{Y} = \widehat{Y}, \quad if \quad Y_2 + v_2 \times t \le \widehat{Y} \le Y_1 - v \times t \tag{10}$$

$$\widehat{Y} = Y, \quad else$$

Here, Y is the real ordinate of the vehicle, $\widehat{Y}$ is the predicted ordinate of the subject vehicle, Y_1 is the ordinate of lead vehicle, Y_2 is the ordinate of lag vehicle, v, v_2 are the velocity of the subject vehicle and lag vehicle, respectively, and t is the minimum safety time headway which equals to 1 s.

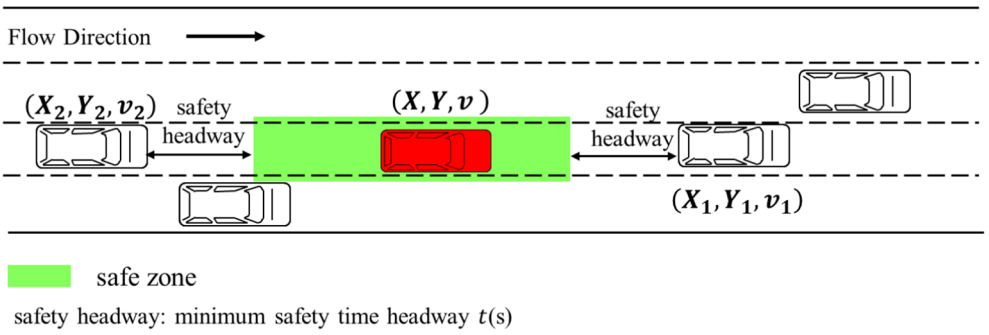


Fig. 9. Safety distance of longitudinal constrains.

2.3.3. HRC LSTM algorithm

As mentioned above, the methods of retraining and longitudinal constrains are integrated into a complete algorithmic process. The algorithm is developed as the following steps.

Algorithm 1 Hybrid Retraining Constrained LSTM

Initialize training dataset D = {($\boldsymbol{x}_1$, $\boldsymbol{y}_1$), ($\boldsymbol{x}_2$, $\boldsymbol{y}_2$), ⋯}, dataset D_1 is half of dataset D, dataset D_2 is another and dataset D_3= ∅
X (D), Y (D) stands for input($\boldsymbol{x}_1$, $\boldsymbol{x}_2$,⋯) and output($\boldsymbol{y}_1$, $\boldsymbol{y}_2$,⋯) of dataset D
Initialize LSTM Neural Network M with training in dataset D_1
for episode = 1, N **do**
  for $\boldsymbol{x}_i$ in X (D_2) **do**
    Prediction $\hat{\boldsymbol{y}}_i$= M ($\boldsymbol{x}_i$) which is the output of LSTM-NN M
    if longitudinal prediction of $\hat{\boldsymbol{y}}_i$ is reasonable **then**
      Store $\hat{\boldsymbol{y}}_i$ in X (D_3)
    else
    Store lateral prediction of $\hat{\boldsymbol{y}}_i$ and corresponding real ordinate in X (D_3)
    end if
  end for
  Set Y (D_3) = Y (D_2)
  A new dataset D* is the aggregation of D_1 and D_3
  A new LSTM-NN, i.e., a HRC LSTM-NN M * is retrained by dataset D*
end for

3. Case study

In this section, we use NGSIM dataset which is commonly used in transportation research to train and test the HRC LSTM model and evaluate the model with a group of systematic indictors from both longitudinal and lateral aspects.

3.1. Data preparation

As a deep learning algorithm, the improved HRC LSTM model requires massive trajectories data including both CF and LC behaviors to be trained. We thus adopt the I-80 dataset from NGSIM dataset to train the integrated model. The vehicles driving on the middle lane (lane 2–5) from 4:00 p.m. to 4:15 p.m. are collected and used. For each vehicle, the positions of the closest preceding and following vehicles within 100 m (since the range of commonly used sensors like LIDAR or RADAR are 60–175 m (Cho et al., 2014)) in the current lane and adjacent lanes are extracted (6 blue vehicles as shown in Fig. 5; if there is no vehicle within 100 m ahead or behind, we consider virtual preceding or following vehicles at 100 m for the sake of calculation convenience). Finally, 1230 vehicles' trajectories are obtained (including 200 LCs and more than 800,000 CF samples), where 2/3 of data are used for training and the rest 1/3 of data are used for testing. On the other hand, although the NGSIM dataset contain all sorts of inconsistencies and errors, the main issues are within the speed and acceleration data which are not used in this study (Montanino and Punzo, 2013, 2015; Treiber and Kesting, 2013). Thus, we believe that the errors of NGSIM data should not impact the results of model.

3.2. Parameters setting

In the LSTM model, there are many parameters that affect the performance of the model. Generally, parameters of DNN are usually selected by experience and few empirical experiment results, as it is very difficult to find the global optimal sets of parameters for any DNN (Christiano et al., 2017; Cornebise et al., 2018; Santoro et al., 2017; Weber et al., 2017). People even hardly perform a systematic grid search to select the optimal parameters owing to the high computational cost. Thus in this paper, we choose the

Table 1
Parameters and their values in HRC LSTM.

Parameter	Description	Value
Input dimension	Dimensions of input layer	14
Output dimension	Dimensions of output layer	2
Batch size	Number of training cases over each Adam update	32
Historical length	Number of past frames (0.1 s)	80
Hidden layer number	Number of LSTM layer	1
Neuron in the hidden layer	Number of neurons in the hidden layer	100
Dropout rate	The rate used in dropout layer	0.2
Training epochs	Number of training update	15
Retraining epochs	Number of retraining update	5
Loss function	The function to calculate loss	MSE
Activation function	The function to activate output	Sigmoid
Optimizer	The function to minimize loss	Adam
Safety time headway	Headway kept for safety (t (s) in Fig. 7)	1

parameters (batch size, hidden layer number, training epochs, etc.) by performing an informal search on the experiments as per previous study (Cornebise et al., 2018; Silver et al., 2017). We conducted sensitivity experiments with different parameter settings to test the HRC LSTM predictive performance for the trajectories. The final parameter settings can be found in Table 1.

3.3. The model performance evaluation indicators

For simultaneously modeling CF and LC behaviors, only evaluating the error of predictive trajectories is not sufficient. An in-depth analysis on the separate lateral and longitudinal behaviors is also needed. Therefore, several evaluation indictors from three perspectives are used in this study: lateral accuracy, longitudinal accuracy and overall accuracy of trajectories, and final result is the average value of all the vehicles.

With respect to the lateral accuracy, the lateral prediction accuracy (the percentage of successful LC predictions of observed LCs) in Eq. (11) shows how accurate the predictive LCs are, where a successful prediction means the same vehicle as in reality makes LC around the same time, and the distributions of accepted LC time gaps between observed data and simulated data are also compared. For longitudinal accuracy, two commonly used metrics, longitudinal error (Ossen and Hoogendoorn, 2011) as shown in Eq. (12) which considers not only vertical coordinates but also speed and mixed gap error which considers gap errors as shown in Eq. (13) (Kesting and Treiber, 2008), are used. Regarding the overall accuracy, the model error in Eq. (14) is an extended version of the longitudinal error, which calculates the error of longitudinal and lateral location and speed between observed data and simulated data. Reasonable prediction accuracy as shown in Eq. (15) (the percentage of reasonable predictions which meet the safety distance condition as shown in Fig. 7 of the total predictions) is another indicator for overall accuracy, which tells the reasonableness of integrated model.

$$lateral\ \ prediction\ \ accuracy = \frac{num\ \ of\ \ successful\ \ LC\ \ predictions}{total\ \ observed\ \ LC\ \ number} \tag{11}$$

$$longitude\ \ error = \frac{\sqrt{\sum_t (y_o - y_s)^2}}{\sqrt{\sum_t y_o^2} + \sqrt{\sum_t y_s^2}} + \frac{1}{2} \times \frac{\sqrt{\sum_t (v_o - v_s)^2}}{\sqrt{\sum_t v_o^2} + \sqrt{\sum_t v_s^2}} \tag{12}$$

$$mixed\ \ gap\ \ error = \sqrt{\frac{\sum_t \frac{(G_o - G_s)^2}{|G_o|}}{\sum_t G_o}} \tag{13}$$

$$model\ \ error = \frac{\sqrt{\sum_t (x_o - x_s)^2}}{\sqrt{\sum_t x_o^2} + \sqrt{\sum_t x_s^2}} + \frac{\sqrt{\sum_t (y_o - y_s)^2}}{\sqrt{\sum_t y_o^2} + \sqrt{\sum_t y_s^2}} + \frac{1}{2} \times \frac{\sqrt{\sum_t (v_o - v_s)^2}}{\sqrt{\sum_t v_o^2} + \sqrt{\sum_t v_s^2}} \tag{14}$$

$$reasonable\ \ prediction\ \ accuracy = \frac{num\ \ of\ \ reasonable\ \ predictions}{total\ \ number} \tag{15}$$

where

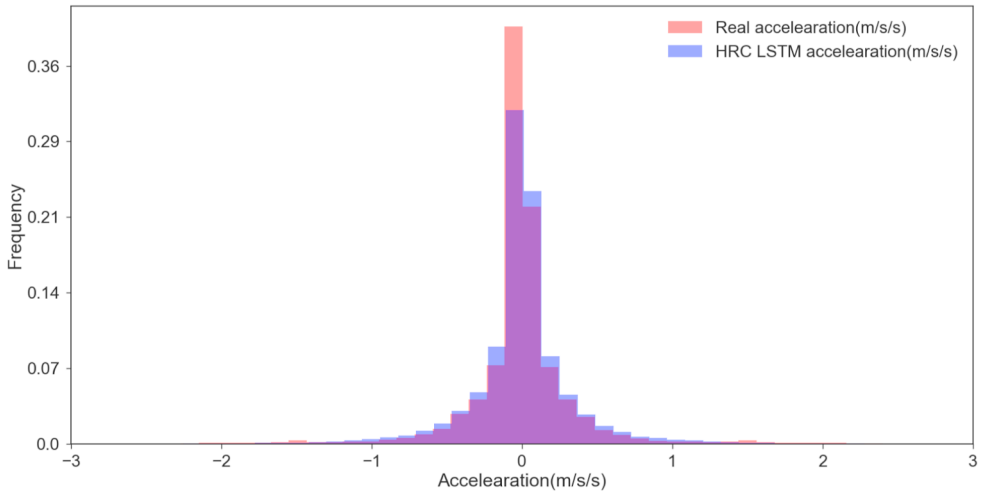
x_o, x_s are local lateral positions (ft) of observed data and simulated data;
y_o, y_s are local longitudinal positions (ft) of observed data and simulated data;
v_o, v_s are speeds (ft/s) of observed data and simulated data;
G_o, G_s are gap (ft) of observed data and simulated data;
t is the frame (0.1 s) number.

Table 2
Comparison results of different LSTM models.

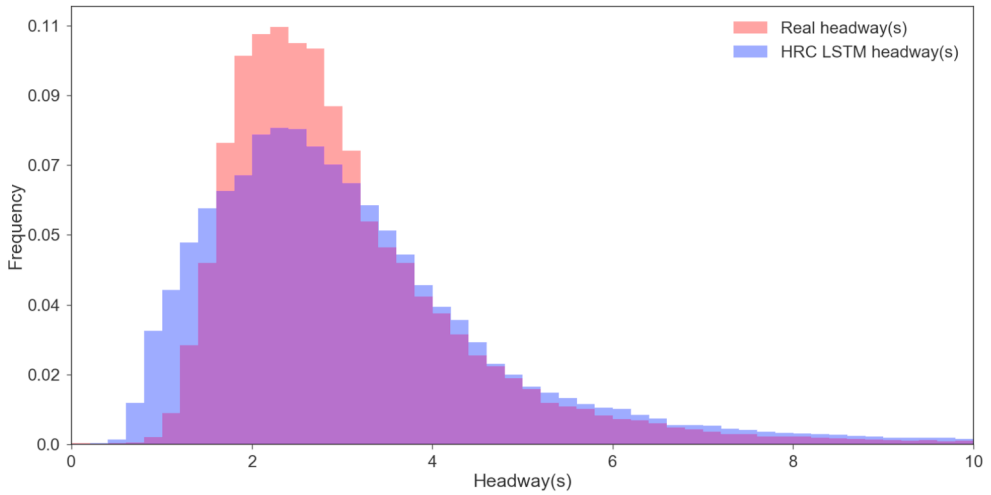
Model		Overall		Longitudinal		Lateral
		Model error	Reasonable prediction accuracy	Longitudinal error	Mixed gap error	Prediction accuracy
LSTM	Training dataset	0.088	100%	0.081	0.90	85%
	Testing dataset	**0.094**	**50%**	**0.075**	**0.90**	**74%**
Retraining LSTM	Training dataset	0.071	99%	0.064	0.44	91%
	Testing dataset	**0.100**	**70%**	**0.080**	**0.77**	**87%**
HRC LSTM	Training dataset	0.059	97%	0.056	0.34	100%
	Testing dataset	**0.049**	**96%**	**0.045**	**0.39**	**100%**

3.4. Results analysis

To evaluate the performance of the HRC LSTM model, training and testing experiments with the parameters presented in Table 1 are conducted with the HRC LSTM model. In addition, the original LSTM and retraining LSTM without longitudinal constrains (shown as retraining LSTM below) are trained with the same data and parameters for comparison as well. The comparison results of different



(a) Histogram of real and simulated acceleration.



(b) Histogram of real and simulated headway.

Fig. 10. Histogram of real and simulated acceleration (a) and headway (b).

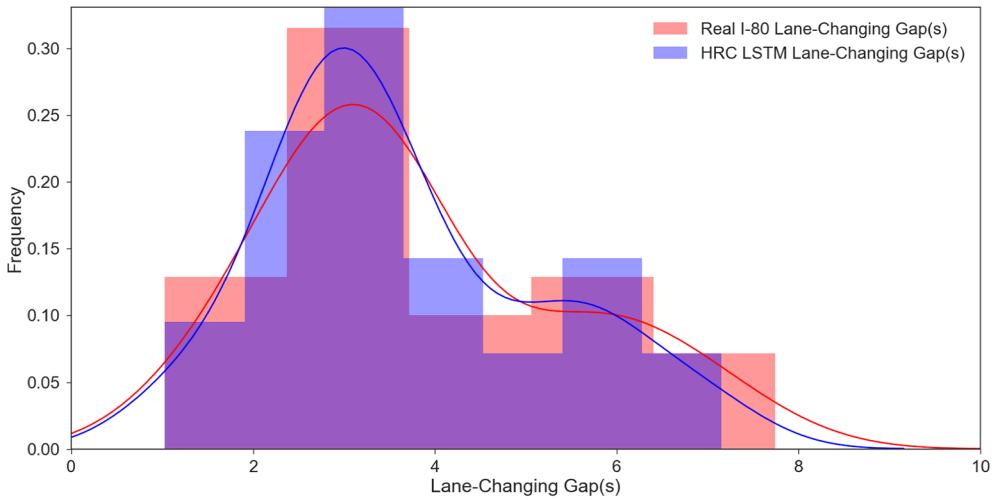


Fig. 11. Time gap distribution of LCs with HRC LSTM model.

models with both the training set and the testing set are presented in Table 2. In general, the performance of models are compared with the testing dataset. The comparison between the training dataset and testing dataset can indicate the ability of models on overfitting issue. In order to systematically assess the performance of the integrated model, the results analysis of three models is divided into several subsections, including overall results, longitudinal results, lateral results and trajectory comparison.

3.4.1. Overall results

As shown in Table 2, the results of LSYM model and retraining LSTM model with testing dataset are slightly worse than that with training dataset, while the HRC LSTM model have even better results with the testing dataset, indicating its better performance on overfitting issue than other two models. For the testing results, from the original LSTM model to the HRC LSTM model, the model error drops from 0.094 to 0.049, while the model error of retraining LSTM model is greater than other two models. However, regarding the reasonable prediction rate, the original LSTM model can only have half reasonable predictions, while the other half predictions do not satisfy the safety distance condition. With retraining, the reasonable prediction rate grows to 70%, and finally in HRC LSTM model, the reasonable prediction rate is improved to 96%.

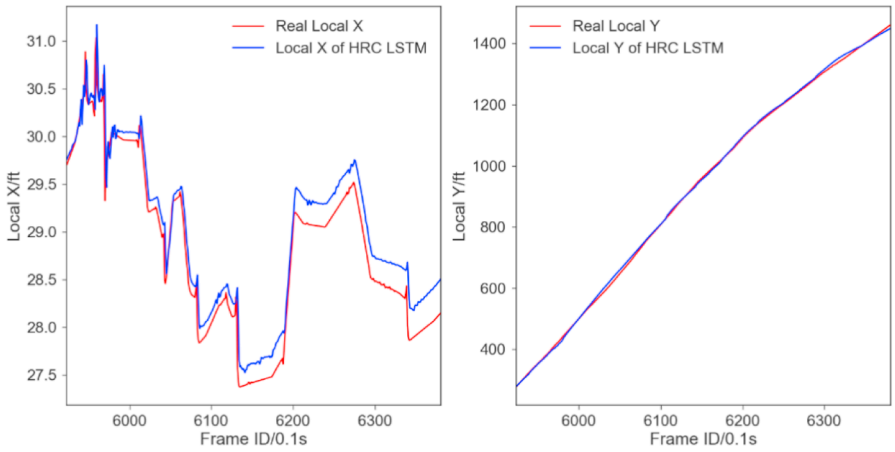
3.4.2. Longitudinal results

Similar to the model error, the longitudinal error and mixed gap error of the HRC LSTM model is the minimum of all the three models. To further evaluate the performance of the integrated model in CF behavior, we compare the longitudinal error with a widely used CF model, intelligent driver model (IDM) (Treiber et al., 2000) because of its well demonstrated capability of reproducing many characteristics of traffic flow. In a typical CF calibration study using IDM, trajectory data collected during peak hours on Dutch Motorway A2 (we cannot find any other IDM using NGSIM dataset and evaluation indicator in Eq. (12) at the same time) is used and a sequential quadratic programming method is applied to calibrate IDM, and longitudinal error of IDM is obtained which ranges from 0.005 to 0.422 (Wang et al., 2010). As we can see, the longitudinal error in HRC LSTM model is in the range of error in IDM, at least HRC LSTM is not definitely inferior to the pure CF model of IDM.

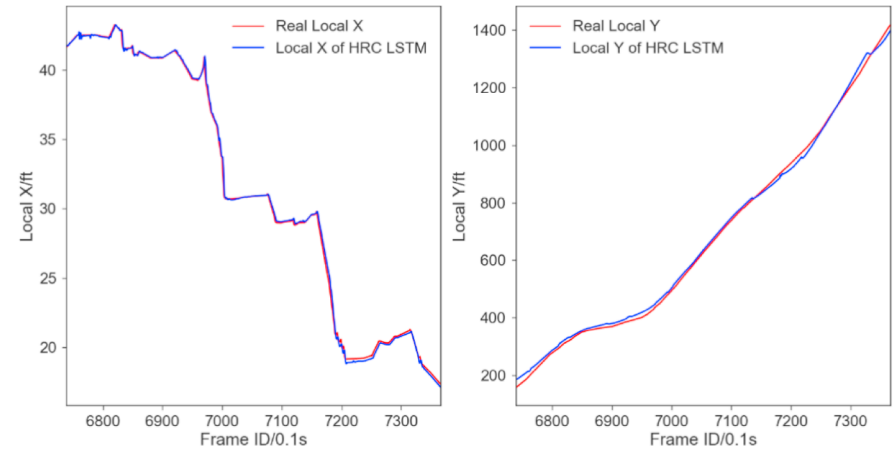
In addition, we have analyzed the distribution of headway and acceleration in terms of microscopic behaviors. As Fig. 10 shows, the distributions of the acceleration and time headway are very consistent between the HRC-LSTM results and the real data. The differences between the distribution are evaluated with Welch's *t*-test which remains robust for skewed distributions (Fagerland, 2012). Finally, the p-values are 0.67 for acceleration and 0.09 for headway, respectively, which are all greater than 0.05. It means there are 95% confidence to assume no significant difference between predictions and real longitudinal behaviors.

3.4.3. Lateral results

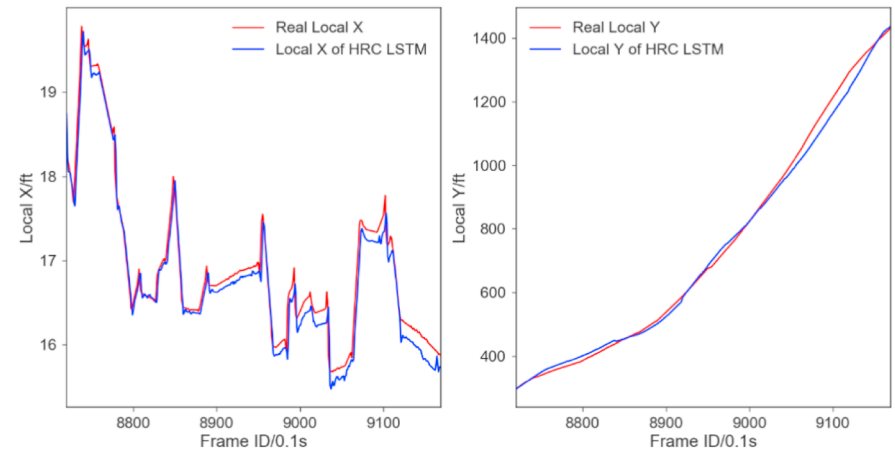
With the testing dataset, the original LSTM model can only predict 42 LCs from the total 57 LCs (lateral prediction accuracy is 74%), and the retraining LSTM model can predict 87% LCs, while HRC LSTM model can accurately predict all the LCs with a 100% successful prediction rate. The high prediction accuracy of HRC LSTM indicates the feasibility of integrated model for LC modeling. In addition, to validate the characteristics of LC behavior, the time gap distributions of LCs reproduced by HRC LSTM model are compared with the real data as shown in Fig. 11. Since the prediction accuracy and reasonable prediction accuracy of the original LSTM and retraining LSTM model are much lower than HRC LSTM, their time gap distributions are not extracted for the comparison. The differences between the distribution reproduced by the integrated models and real data are also evaluated with Welch's *t*-test. Finally, the p-value is 0.32, which is all much greater than 0.05. It means there are 95% confidence to assume no significant difference between predictions and real LCs. The results show that HRC LSTM model can not only predict each LC, but also decently depict the LC characteristics. On the other hand, it reflects that the longitudinal constrains promote the CF modeling significantly.



(a) Lateral movement in one lane of vehicle 1925 (from training dataset).



(b) Continuous lane changing and stop and go driving of vehicle 2285 (from training dataset).

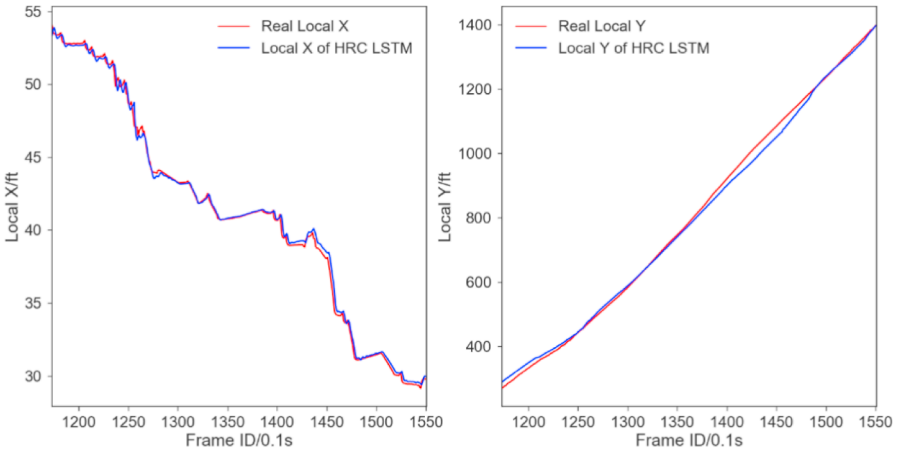


(c) Lateral movement in one lane of vehicle 3125 (from testing dataset).

Fig. 12. Comparison between observed and simulated trajectories.

3.4.4. Individual comparisons

To show the results in terms of every single vehicle, we plot the one-to-one comparison by selecting several typical trajectories from training dataset and testing dataset, respectively, as shown in Fig. 12: (a) lateral movement in one lane (from training dataset), (b) continuous lane changing and stop and go driving (from training dataset), (c) lateral movement in one lane (from testing dataset),



(d) Continuous lane changing of vehicle 3344 (from testing dataset).

Fig. 12. (*continued*)

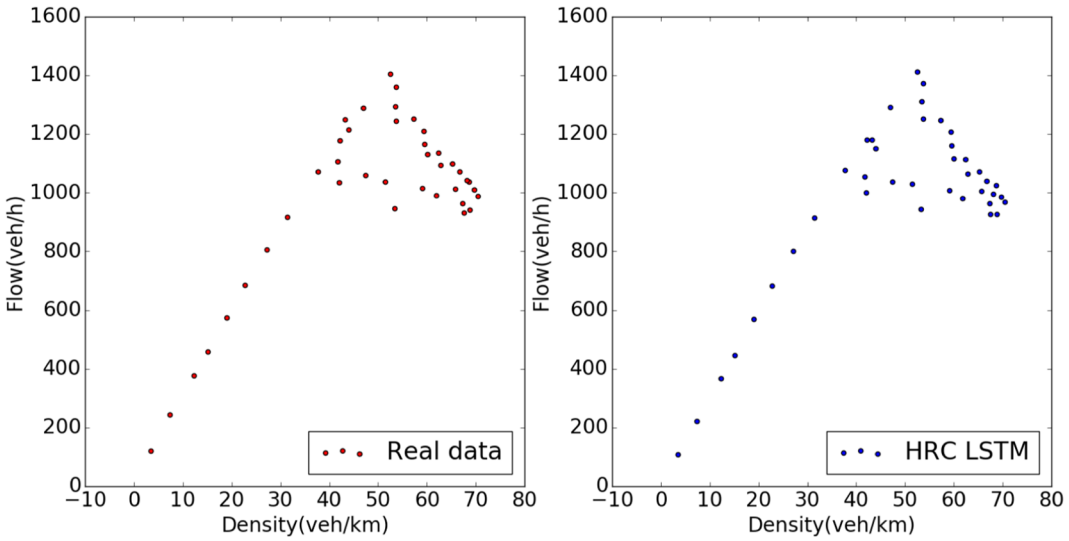


Fig. 13. Flow-density comparison between real data and simulation results.

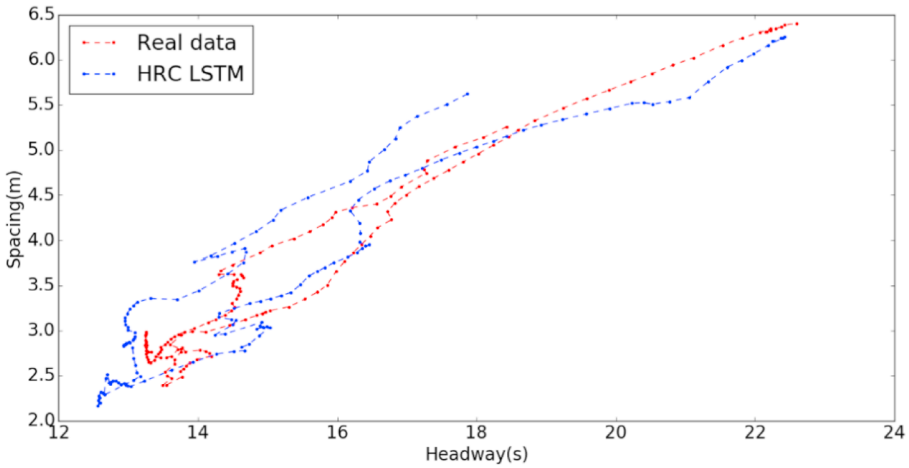
(d) continuous lane changing (from testing dataset). The results illustrate that the simulated trajectories are much consistent with the observed ones and can capture both the longitudinal and lateral movements of vehicles, which further prove the good performance of the proposed model.

3.4.5. Congestion dynamics comparisons

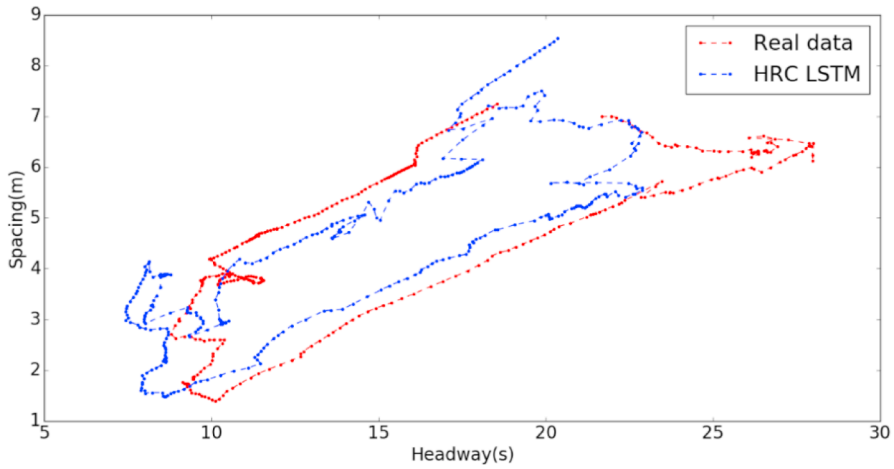
The ability of re-capturing macro/micro congestion dynamics is critical for microscopic behavioral models to validate their performance. Therefore, we further analyze the corresponding flow/density and spacing/time headway characteristics of the simulation vehicles in testing dataset.

As for the flow/density characteristics, we extract the real and simulated data of lane 2 on I-80 for further analysis. Please note that this lane is only selected for validation and we can obtain similar conclusion from other data. We estimate the flow rate and density of real data and simulation results with Edie's definition (Edie, 1963), where the time aggregation interval is one minute and the location is 400 ft of local Y (Lu et al., 2009). The comparison of flow-density diagrams is shown in Fig. 13. The trend of flow-density of simulation results is basically consistent with that of real data, which is the well-known quadric curve.

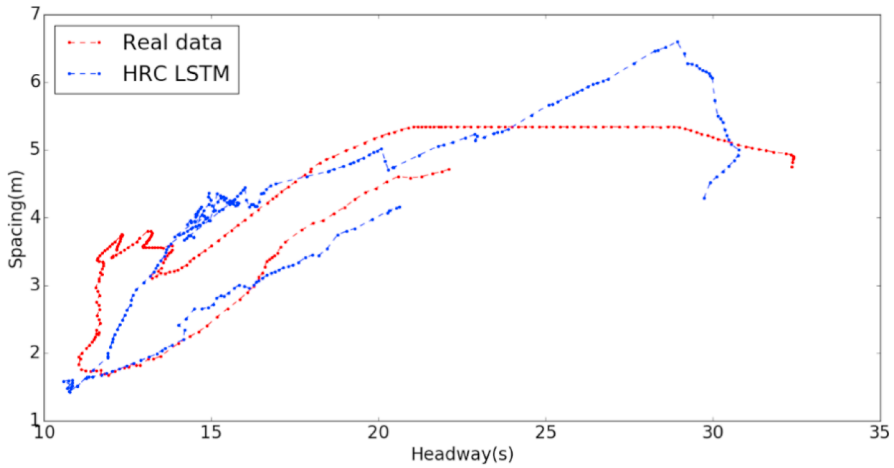
For the relationship between spacing and time headway of simulation results, as shown in Fig. 14, HRC LSTM can reproduce the hysteresis loop which is well-known in the real traffic (Zhang, 1999). This further demonstrates the capability of proposed model in generating typical features of real traffic.



(a) Spacing-headway of vehicle 2938.



(b) Spacing-headway of vehicle 2940.



(c) Spacing-headway of vehicle 2962.

Fig. 14. Spacing-time headway comparison between real data and simulation results.

Table 3
Comparison results of different testing datasets.

Dataset	Overall		Longitudinal		Lateral
	Model error	Reasonable prediction accuracy	Longitudinal error	Mixed gap error	Prediction accuracy
I-80	0.049	96%	0.045	0.39	100%
US-101	0.036	99%	0.033	0.39	100%

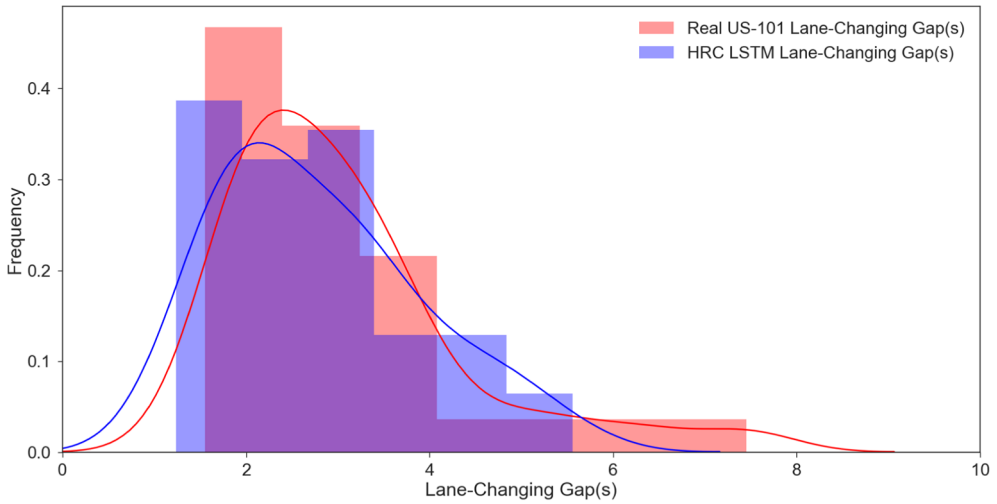


Fig. 15. Time gap distribution of LCs with HRC LSTM model and US-101 dataset.

4. Transferability analysis

To further examine the applicability of the developed HRC LSTM model based on the I-80 dataset on other freeways, we evaluate the transferability of the proposed model by directly applying this model with the US-101 dataset in NGSIM dataset. The trajectories of vehicles driving on the median three lanes (lane 1, 2, 3) and corresponding surrounding vehicles' positions information are collected. Finally, 373 vehicles' trajectories are obtained, including 35 LCs and more than 240,000 samples. The results of the HRC LSTM model with different testing datasets (i.e. the original I-80 dataset and the US-101 dataset) are compared as shown in Table 3.

The HRC LSTM model performs quite well with the US-101 dataset with even better results than that of the original testing dataset Moreover, the time gap distribution of LCs is proved to be almost same with the real data by Welch's *t*-test as shown in Fig. 15. The P-value is 0.21 larger than 0.05 which indicates that there is no significant difference between the two distributions. Thus the high consistency between simulated and real data illustrates the transferability of the developed model.

5. Conclusions

CF and LC behaviors are two basic types of vehicle movements which are generally modeled separately in the literature. In this study, we proposed HRC LSTM model to model the two basic behaviors simultaneously, so that the vehicle can realize fully autonomous motion simulation on the freeway. With only the positions information of the surrounding vehicles, the proposed model can predict the trajectories of vehicles. The I-80 dataset in NGSIM dataset is then used to train and test the improved HRC LSTM model, while the US101 dataset is used to evaluate the transferability of the model. The main conclusions are as follows:

(1) The improved HRC LSTM model can model realistic driving behavior and show the ability of re-capturing macro/micro congestion dynamics. The accuracy of longitudinal trajectories of this model is close to that of IDM, while the prediction accuracy of LC is 100% and the LC time gap distribution characteristics are not statistically significantly difference with the real LC behaviors. And the trend of flow-density of simulation results is basically consistent with that of real data, meanwhile the hysteresis loop is also reproduced.
(2) With the novel training approach, the HRC LSTM model outperforms the retraining LSTM model and the original LSTM model in terms of all the indicators, which further demonstrates the necessary of additive retraining process and longitudinal constrains.
(3) When directly implementing the model developed with I-80 dataset to the US-101 dataset, the prediction performance is still very high, while the overall model error is smaller than that with the original testing dataset, and which shows the HRC LSTM has very good transferability.

In summary, the feasibility of using deep learning method for modeling CF and LC behaviors at the same time is validated in this paper. Particularly, considering the development of automated vehicles, the proposed model can be used effectively for the intelligent decision making and motion planning of automated vehicles. Based on the simultaneous modeling, future research could study how the interaction between CF and LC behaviors impact the traffic flow characteristics, such as the formation and development of traffic oscillation. On the other hand, though the HRC LSTM which can reproduce trajectory data is validated, but the translation of the adopted approach into a robust driver behavior model, that can capture different driving conditions with limited set of trajectories, is still a subject a research. Given the nature of the neural network, the adopted approach is more data dependent than other behavioral/physics based traffic models and requires extra care in calibrating and validating the corresponding parameters; these parameters are not interpretable in terms of physical and behavioral characteristics of the vehicles/drivers.

Acknowledgement

We would like to thank all the reviewers for their insightful comments and suggestions. And this research is sponsored by the National Key Research and Development Program of China (2018YFB1600505), the National Natural Science Foundation of China (U1764261), and the “Shuguang Program” supported by Shanghai Education Development Foundation and Shanghai Municipal Education Commission (18SG21).

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.trc.2019.05.021.

References

Ahmed, K.I., 1999. Modeling Drivers' Acceleration and Lane Changing Behavior. Massachusetts Institute of Technology.
Ben Taieb, S., Bontempi, G., Atiya, A.F., Sorjamaa, A., 2012. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. Expert Syst. Appl. https://doi.org/10.1016/j.eswa.2012.01.039.
Blum, A., Mitchell, T., 1998. Combining labeled and unlabeled data with co-training, in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory - COLT' 98. pp. 92–100. https://doi.org/10.1145/279943.279962.
Brackstone, M., McDonald, M., 1999. Car-following: a historical review. Transp. Res. Part F Traffic Psychol. Behav. 2, 181–196. https://doi.org/10.1016/S1369-8478(00)00005-X.
Cho, H., Seo, Y.W., Kumar, B.V.K.V., Rajkumar, R.R., 2014. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In: Proc. – IEEE Int. Conf. Robot. Autom, pp. 1836–1843.
Christiano, P., Leike, J., Brown, T.B., Martic, M., Legg, S., Amodei, D., 2017. Deep reinforcement learning from human preferences.
Cornebise, J., Keane, P.A., Ronneberger, O., 2018. Clinically applicable deep learning for diagnosis and referral in retinal disease. Nat. Med. https://doi.org/10.1038/s41591-018-0107-6.
Cosgun, A., Ma, L., Chiu, J., Huang, J., Demir, M., Anon, A.M., Lian, T., Tafish, H., Al-Stouhi, S., 2017. Towards full automated drive in urban environments: a demonstration in GoMentum Station, California. In: 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE, pp. 1811–1818.
Djuric, N., Radosavljevic, V., Cui, H., Nguyen, T., Chou, F.-C., Lin, T.-H., Schneider, J., 2018. Short-term Motion Prediction of Traffic Actors for Autonomous Driving using Deep Convolutional Networks. eprint arXiv: 1808.05819.
Edie, L., 1963. Discussion of traffic stream measurements and definitions. Port New York Auth.
Fagerland, M.W., 2012. T-tests, non-parametric tests, and large studiesa paradox of statistical practice? BMC Med. Res. Methodol. 12. https://doi.org/10.1186/1471-2288-12-78.
Gers, F.A., Schmidhuber, J., Cummins, F., 2000. Learning to forget: continual prediction with LSTM. Neural Comput. 12, 2451–2471. https://doi.org/10.1162/089976600300015015.
Hidas, P., 2002. Modelling lane changing and merging in microscopic traffic simulation. Transp. Res. Part C Emerg. Technol. 10, 351–371. https://doi.org/10.1016/S0968-090X(02)00026-8.
Hidas, P., 2005. Modelling vehicle interactions in microscopic simulation of merging and weaving. Transp. Res. Part C Emerg. Technol. 13, 37–62. https://doi.org/10.1016/J.TRC.2004.12.003.
Hochreiter, S., Urgen Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
Huang, X., Sun, J., Sun, J., 2018. A car-following model considering asymmetric driving behavior based on long short-term memory neural networks. Transp. Res. Part C Emerg. Technol. https://doi.org/10.1016/j.trc.2018.07.022.
Hunt, J.G., Lyons, G.D., 1994. Modelling dual carriageway lane changing using neural networks. Transp. Res. Part C 2, 231–245. https://doi.org/10.1016/0968-090X(94)90012-4.
Janocha, K., Czarnecki, W.M., 2017. On Loss Functions for Deep Neural Networks in Classification 25, 49–59. https://doi.org/10.4467/20838476SI.16.004.6185.
Kesting, A., Treiber, M., 2008. Calibrating Car-Following Models using Trajectory Data: Methodological Study 1–17. https://doi.org/10.3141/2088-16.
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 Cybern. Part A Syst. Humans 42, 1440–1449. https://doi.org/10.1109/TSMCA.2012.2192262.
Kingma, D.P., Ba, J.L., 2015. Adam: a method for stochastic optimization. Int. Conf. Learn. Represent. 2015, 1–15. https://doi.org/http://doi.acm.org.ezproxy.lib.ucf.edu/10.1145/1830483.1830503.
Kita, H., 1999. A merging–giveway interaction model of cars in a merging section: a game theoretic analysis. Transp. Res. Part A Policy Pract. 33, 305–312. https://doi.org/10.1016/S0965-8564(98)00039-1.
Kita, H., Tanimoto, K., Fukuyama, K., 2002. A game theoretical analysis of merging-giveway interaction: a joint estimation model. Transp. Traffic Theory 15, 503–518.
Laval, J.A., Daganzo, C.F., 2006. Lane-changing in traffic streams. Transp. Res. Part B Methodol. 40, 251–264. https://doi.org/10.1016/j.trb.2005.04.003.
Li, J., Dai, B., Li, X., Li, C., Di, Y., 2017. A real-time and predictive trajectory-generation motion planner for autonomous ground vehicles. In: 2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC). IEEE, pp. 108–113.
Li, L., Lv, C., Cao, D., Zhang, J., 2018. Retrieving common discretionary lane changing characteristics from trajectories. IEEE Trans. Veh. Technol. https://doi.org/10.1109/TVT.2017.2771144.
Li, M., Zhou, Z., 2005. SETRED: self-training with editing. Adv. Knowl. Discov. Data Min. 611–621. https://doi.org/10.1007/11430919_71.
Lu, X.Y., Varaiya, P., Horowitz, R., 2009. Fundamental Diagram modelling and analysis based NGSIM data. IFAC Proceedings Volumes (IFAC-PapersOnline).
Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y., 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp. Res. Part C Emerg. Technol. 54, 187–197. https://doi.org/10.1016/j.trc.2015.03.014.
Montanino, M., Punzo, V., 2013. Making NGSIM data usable for studies on traffic flow theory. Transp. Res. Rec. J. Transp. Res. Board 2390, 99–111. https://doi.org/

10.3141/2390-11.
Montanino, M., Punzo, V., 2015. Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. Transp. Res. Part B Methodol. https://doi.org/10.1016/j.trb.2015.06.010.
Moridpour, S., Rose, G., Sarvi, M., 2009. Modelling the heavy vehicle drivers' lane changing decision under heavy traffic conditions. Road Transp. Res.
Ossen, S., Hoogendoorn, S.P., 2011. Heterogeneity in car-following behavior: theory and empirics. Transp. Res. Part C Emerg. Technol. 19, 182–195. https://doi.org/10.1016/j.trc.2010.05.006.
Panwai, S., Dia, H., 2007. Neural agent car-following models. IEEE Trans. Intell. Transp. Syst. 60–70. https://doi.org/10.1109/TITS.2006.884616.
Saifuzzaman, M., Zheng, Z., 2014. Incorporating human-factors in car-following models: a review of recent developments and research needs. Transp. Res. Part C Emerg. Technol. 48. https://doi.org/10.1016/j.trc.2014.09.008.
Santoro, A., Raposo, D., Barrett, D.G.T., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T., 2017. A simple neural network module for relational reasoning 1–16. https://doi.org/10.1371/journal.pone.0044251.
Schmidhuber, J., 2015. Deep Learning in neural networks: an overview. Neural Netw. https://doi.org/10.1016/j.neunet.2014.09.003.
Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., Van Den Driessche, G., Graepel, T., Hassabis, D., 2017. Mastering the game of Go without human knowledge. Nature 550, 354–359. https://doi.org/10.1038/nature24270.
Sun, J., Ouyang, J., Yang, J., 2014a. Modeling and analysis of merging behavior at expressway on-ramp bottlenecks. Transp. Res. Rec. J. Transp. Res. Board 2421, 74–81. https://doi.org/10.3141/2421-09.
Sun, J., Zhao, L., Zhang, H., 2014b. Mechanism of early-onset breakdown at on-ramp bottlenecks on Shanghai, China, expressways. Transp. Res. Rec. J. Transp. Res. Board 2421, 64–73. https://doi.org/10.3141/2421-08.
Talebpour, A., Mahmassani, H.S., Hamdar, S.H., 2015. Modeling lane-changing behavior in a connected environment: a game theory approach. Transp. Res.Procedia. 420–440. https://doi.org/10.1016/j.trpro.2015.06.022.
Toledo, T., 2007. Driving behaviour: models and challenges. Transp. Rev. https://doi.org/10.1080/01441640600823940.
Toledo, T., Koutsopoulos, H.N., Ben-Akiva, M., 2007. Integrated driving behavior modeling. Transp. Res. Part C Emerg. Technol. 15, 96–112. https://doi.org/10.1016/j.trc.2007.02.002.
Tomar, R.S., Verma, S., Tomar, G.S., 2010. Prediction of lane change trajectories through neural network. In: Proceedings - 2010 International Conference on Computational Intelligence and Communication Networks, CICN 2010, pp. 249–253.
Treiber, M., Hennecke, A., Helbing, D., 2000. Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E Stat. Phys. Plasmas, Fluids Relat. Interdiscip. Top. 62, 1805–1824. https://doi.org/10.1103/PhysRevE.62.1805.
Treiber, M., Kesting, A., 2013. Microscopic calibration and validation of car-following models – a systematic approach. Procedia Soc. Behav. Sci. 80, 922–939. https://doi.org/10.1016/J.SBSPRO.2013.05.050.
Vogel, K., 2003. A comparison of headway and time to collision as safety indicators. Accid. Anal. Prev. 35, 427–433. https://doi.org/10.1016/S0001-4575(02)00022-2.
Wang, H., Wang, W., Chen, J., Jing, M., 2010. Using trajectory data to analyze intradriver heterogeneity in car-following. Transp. Res. Rec. J. Transp. Res. Board 2188, 85–95. https://doi.org/10.3141/2188-10.
Wang, Q., Li, Z., Li, L., 2014. Investigation of discretionary lane-change characteristics using next-generation simulation data sets. J. Intell. Transp. Syst. Technol. Planning, Oper. https://doi.org/10.1080/15472450.2013.810994.
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. https://doi.org/10.1109/TITS.2017.2706963.
Waymo, 2018. On the road to Fully Self-driving - Waymo Safety Report, Waymo Safety Report.
Weber, T., Racanière, S., Reichert, D.P., Buesing, L., Guez, A., Rezende, D.J., Badia, A.P., Vinyals, O., Heess, N., Li, Y., Pascanu, R., Battaglia, P., Hassabis, D., Silver, D., Wierstra, D., 2017. Imagination-Augmented Agents for Deep Reinforcement Learning. https://doi.org/10.1002/aic.690170452.
Wei, D., Liu, H., 2013. Analysis of asymmetric driving behavior using a self-learning approach. Transp. Res. Part B Methodol. 47, 1–14. https://doi.org/10.1016/j.trb.2012.09.003.
Wilson, D.L., 1972. Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. Syst. Man Cybern. 2, 408–421. https://doi.org/10.1109/TSMC.1972.4309137.
Wollmer, M., Blaschke, C., Schindl, T., Schuller, B., Farber, B., Mayer, S., Trefflich, B., 2011. Online driver distraction detection using long short-term memory. Intell. Transp. Syst. IEEE Trans. 12, 574–582. https://doi.org/10.1109/TITS.2011.2119483.
Woo, H., Ji, Y., Kono, H., Tamura, Y., Kuroda, Y., Sugano, T., Yamamoto, Y., Yamashita, A., Asama, H., 2017. Lane-change detection based on vehicle-trajectory prediction. IEEE Robot. Autom. Lett. 2, 1109–1116. https://doi.org/10.1109/LRA.2017.2660543.
Wu, J., Brackstone, M., McDonald, M., 2000. Fuzzy sets and systems for a motorway microscopic simulation model. Fuzzy Sets Syst. 116, 65–76. https://doi.org/10.1016/S0165-0114(99)00038-X.
Zhang, H.M., 1999. A mathematical theory of traffic hysteresis. Transp. Res. Part B Methodol. https://doi.org/10.1016/S0191-2615(98)00022-8.
Zhang, R., Cao, L., Bao, S., Tan, J., 2017. A method for connected vehicle trajectory prediction and collision warning algorithm based on V2V communication. Int. J. Crashworthiness 22, 15–25. https://doi.org/10.1080/13588265.2016.1215584.
Zheng, J., Suzuki, K., Fujita, M., 2013. Car-following behavior with instantaneous driver-vehicle reaction delay: a neural-network-based methodology. Transp. Res. Part C Emerg. Technol. 36, 339–351. https://doi.org/10.1016/j.trc.2013.09.010.
Zheng, Z., 2014. Recent developments and research needs in modeling lane changing. Transp. Res. Part B Methodol. 60, 16–32. https://doi.org/10.1016/j.trb.2013.11.009.
Zhou, Zhi-Hua, Li, Ming, 2005. Tri-training: exploiting unlabeled data using three classifiers. IEEE Trans. Knowl. Data Eng. 17, 1529–1541. https://doi.org/10.1109/TKDE.2005.186.
Zhou, Z.H., Li, M., 2010. Semi-supervised learning by disagreement. Knowl. Inf. Syst. 24, 415–439. https://doi.org/10.1007/s10115-009-0209-z.
Zhou, M., Qu, X., Li, X., 2017. A recurrent neural network based microscopic car following model to predict traffic oscillation. Transp. Res. Part C Emerg. Technol. 84, 245–264. https://doi.org/10.1016/j.trc.2017.08.027.