Physica A 599 (2022) 127303

Contents lists available at ScienceDirect

Physica A

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

An integrated car-following and lane changing vehicle trajectory prediction algorithm based on a deep neural network

Kunsong Shi [a], Yuankai Wu [b], Haotian Shi [a], Yang Zhou [a,*], Bin Ran [a]

[a] Department of Civil and Environmental Engineering, University of Wisconsin-Madison, 1415 Engineering Drive, Madison, 53706, Wisconsin, United States
[b] Department of Civil Engineering and Applied Mechanics, McGill University, 817 Sherbrooke St W, Montreal, H3A 0C3, Quebec, Canada

ARTICLE INFO

Article history:
Received 30 October 2021
Received in revised form 17 March 2022
Available online 5 April 2022

Keywords:
Vehicle trajectory prediction
Car following
Lane changing
Integrated framework
Neural network with a switch structure

ABSTRACT

Vehicle trajectory prediction is essential for the operation safety and control efficiency of automated driving. Prevailing studies predict car following and lane change processes in a separate manner, ignoring the dependencies of these two behaviors. To remedy this issue, this paper proposes an integrated deep learning-based two-dimension trajectory prediction model that can predict combined behaviors. Specifically, we designed a switch neural network structure based on the attention mechanism, bi-directional long-short term memory (BiLSTM) and Temporal convolution neural network (TCN) to mimic and predict the joint behaviors. Experiments are conducted based on the Next Generation Simulation (NGSIM) dataset to validate the effectiveness of our proposed model. As results indicate, our proposed model outperforms the state-of-art trajectory prediction models and can provide accurate short-term and long-term predictions.

1. Introduction

With fast development of automated vehicle technology, prediction serve as an essential module for automated vehicles. Among prediction targets, accurate vehicle trajectory prediction is a key factor related to the safety and control efficiency of automated driving systems [1]. Many state-of-the-art control models use surrounding vehicle trajectory information to make key decisions and control automated vehicles [2–9]. Inaccurate trajectory information can cause these control methods to become inefficient and unsafe. Therefore, accurate vehicle trajectory prediction is critical to the efficiency and safety of autonomous vehicles.

Due to the importance, many existing studies have paid attention to vehicle trajectory prediction. Among those methods, there are two main types of vehicle movements of interest: lane change movements [10,11] and car following movements [12,13]. The lane change movement is defined as the vehicle moving from one lane to a nearby lane, and the car-following movement is defined as the vehicle continuously following the leading vehicle in the same lane. Furthermore, based on methodology differences, these research can be further divided into two categories: model based methods, and data-driven methods. Model based vehicle trajectory prediction methods predict vehicle movement by a presumed model, usually from traffic flow theory perspective, whereas, the data-driven method predicts vehicle movement in a data-driven fashion, for example using neural networks. The model based prediction methods have great

* Corresponding author.
E-mail address: zhou295@wisc.edu (Y. Zhou).

https://doi.org/10.1016/j.physa.2022.127303
0378-4371/

interpretability to explain certain traffic phenomenons. For example, for car following models, Treiber et al. [14] presented the intelligent driver model (IDM), which relates to using the desired spacing as a parameter in the car-following model. Li et al. [15] proposed a systematic non linear car-following model that can reduce observed traffic oscillation propagation phenomenon. Chen et al. [16] developed an asymmetric behavior car-following model that is developed based on a statistical analysis of driver behaviors. For lane change models, Gipps et al. [17] presented a classical rule based model that was based on a series of specific lane change rules. Hidas et al. [18] proposed an improved rule-based lane change model by categorizing the lane change process into three different categories, including forced, cooperative, and free lane change. Kesting et al. [19] developed a lane change model for the IDM car-following model, which is called the MOBIL model. Laval et al. [20] presented a rule based lane change model using lane-specific macroscopic quantities to model different lane change traffic conditions. However, these model based methods can also introduce prediction errors when a model mismatch occurs, and driver behavior changes drastically.

With recent developments in deep learning, data-driven methods have been favored increasingly, which predict trajectories using only data itself. These methods can circumvent model mismatch caused by assuming a specific model and better capture the temporal-varying behaviors by the extraordinary ability of deep neural networks to learn patterns and features from a large amount of data. Many studies [21–23] have demonstrated promising results utilizing data-driven based methods compared to the traditional model based methods. For car following model, Zhou et al. [24] developed a data-driven car-following model that can effectively capture traffic oscillation in the trajectory using a recurrent neural network. Zhang et al. [25] created a data-driven car-following model using deep long-short term memory(LSTM) that achieved a good prediction performance. Zhao et al. [26] presented generative adversarial network based vehicle trajectory prediction that can capture the behavior of vehicle drivers and achieved great prediction accuracy result. Ma et al. [27] presented a sequence to sequence based data-driven car-following model that can accurately capture heterogeneous driving behaviors. On the other hand, for lane changing models, Xie et al. [28] created a data-driven lane change trajectory prediction model using LSTM, whose results suggest to be a successful LSTM application. Based on that, by introducing two types of attention mechanisms, Scheel et al. [29] achieves better prediction accuracy compared with baseline LSTM models. Differed from the recurrent neural network(RNN)-type methods, Gao et al. [30] proposed a new grouped convolution neural network for prediction lane changing behavior.

Though applied successfully, above mentioned data-driven models largely treat lane change and car following as an independent process, whereas, in widely analyzed traffic flow areas, the dependencies between two movements are significant. On the one hand, the car-following movement can cause instability in the traffic flow, which may affect discretionary lane change. Studies have discovered that different car-following combinations can cause interference in the stability of traffic flow [31]. This instability can cause variations in the possible gap choice of the discretionary lane change. On the other hand, the lane change movement can cause a sudden change in the car following distance and speed difference, which negatively affects the entire traffic flow. Several studies have shown that the lane change movement is the major source of traffic oscillations [32], and causes a 'phantom' congestion [33] on freeways. Moreover, the study [34] has discovered two transition periods in the lane change process, the anticipation period and relaxation period, which can directly impact the proceeding vehicle trajectory and the immediate following vehicle trajectory.

To incorporate the inter-dependencies mentioned above, an integrated vehicle trajectory prediction model is needed. However, difficulties arise since the model needs to not only focus on the continuous longitudinal movement on the current lane but also the discrete lane change choice on the adjacent lanes. Further, the two processes are nested due to the dependencies mentioned above. Hence, an integrated specially designed neural network, which is capable of describing the process and conveying the physical meanings, is more desired. To address the problems mentioned above, we propose a deep learning based integrated two dimensional trajectory prediction model, which is a unified model, to capture the inter-dependency and while exploiting the merits of data-driven methods. Specifically, we designed a deep long short term memory neural network with a switch structure. The switch structure is a module that can generate a lane change prediction output which then can be used directly as a feature to describe the corresponding change of proceeding-following car following relationship. Furthermore, a temporal convolution neural network(TCN) [35] layer is used in our model to further give the model additional temporal depths for better prediction based on previous LSTM algorithms. Additionally, the study focus on discretionary lane change because of the availability of data from all considered lanes.

The rest of the paper is organized as below: Section 2 gives the mathematical problem formulation of our problem; Section 3 gives the structure of our specifically designed neural network structure, and Section 4 delivers the experiment and result. Last but not least, Section 5 gives a conclusion and points out the future work.

2. Mathematical formulation of integrated two-dimensional vehicle trajectory prediction

Vehicle trajectory is described by a Cartesian Coordinate denoted by X,Y, which represents the position of the target vehicle. Our prediction algorithm mainly focused on predicting the car following trajectories longitudinally in Y-axis, and meanwhile predicting the lane change decision described by X-axis. Specifically, at each time point t, we aim to predict the future portfolio of our target vehicle from time t to $t+n$ utilizing the historical information of the target vehicle and the surrounding vehicles from time t to $t-m$. n and m are the prediction horizon and the memory horizon, respectively. To systematically consider the surrounding impact upon the target vehicle, all kinematic states of nine vehicles, including the target vehicle and eight surrounding vehicles, are considered shown as Fig. 1. In Fig. 1, $N_{2,1}$ and $N_{2,3}$ denote the preceding

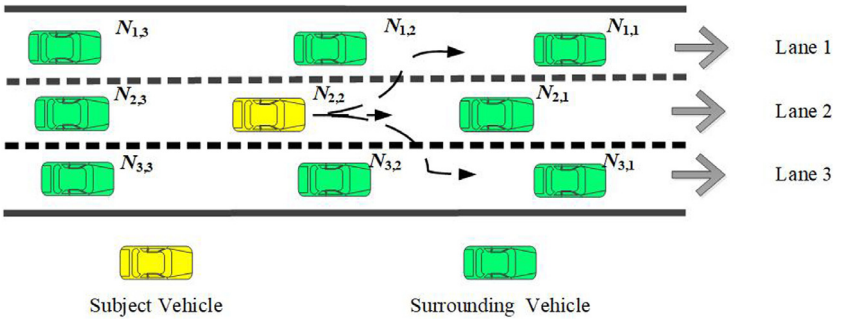


Fig. 1. Schematic diagram of the vehicle trajectory prediction scenario.

vehicle and the following vehicle on the current lane, respectively. $N_{3,1}$, $N_{3,2}$, $N_{3,3}$ and $N_{1,1}$, $N_{1,2}$, $N_{1,3}$ represents vehicles ranked from downstream to upstream on the right and left lane.

To mathematically describe our algorithm, we define the input features for the model as: $\Delta\mathbb{P}_t = [P_{t-m}, P_{t-m+1}, \ldots, P_t]$, $\Delta\mathbb{P}_t \in \mathbb{R}^{6\times3\times m}$, whose element is $p_{i,j,t}$, where i denote lane number and j denotes vehicle number and $p_{i,j,t} = (x_{i,j,t}, y_{i,j,t})$, is the local coordinates of vehicle $N_{i,j}$ at time t; $\mathbb{V}_t = [V_{t-m}, V_{t-m+1}, \ldots, V_t]$, $\mathbb{V}_t \in \mathbb{R}^{3\times3\times m}$, whose element is $v_{i,j,t}$,where i denote to lane number and j denotes the vehicle number, is the speed of vehicle $N_{i,j}$ at time t; $\mathbb{A}_t = [A_{t-m}, A_{t-m+1}, \ldots, A_t]$, $\mathbb{A}_t \in \mathbb{R}^{3\times3\times m}$, whose element is $a_{i,j,t}$, where i denote to lane number and j denotes the vehicle number, is the acceleration of vehicle $N_{i,j}$ at time t; $\Delta\mathbb{V}_t = [\Delta V_{t-m}, \Delta V_{t-m+1} \ldots \Delta V_t]$, $\Delta\mathbb{V}_t \in \mathbb{R}^{3\times3\times m}$, whose element Δv_t is the speed difference between the current vehicle and the preceding vehicle.

In the coordinate system, the current position of the subject vehicle at time t is defined as the referenced origin. The output of our model at time point t is defined as the predicted longitudinal trajectory $Y_t = [P_{2+\phi_t,2,t+1}, P_{2+\phi_t,2,t+2}, \ldots, P_{2+\phi_t,2,t+n}]$, $Y_t \in \mathbb{R}^{2\times n}$, where $P_{2+\phi_t,2,t} = (x_{2+\phi_t,2,t}, y_{2+\phi_t,2,t})$ and predicted lane change status $\phi_t \in \{-1, 0, 1\}$, where -1 is left lane change, 0 is no lane change, 1 is right lane change. The integrated trajectory prediction model is formulated mathematically as:

$$\overrightarrow{\boldsymbol{\phi}}_{\boldsymbol{t}} = H(\Delta\mathbb{P}_t, \mathbb{V}_t, \Delta\mathbb{V}_t, \mathbb{A}_t), \tag{1}$$

$$\Delta v_t = v_{2+\phi_t,2,t} - v_{2+\phi_t,1,t} \tag{2}$$

$$Y_t = G([\overrightarrow{\phi}_t \; S_t]), \tag{3}$$

where $S_t = [\Delta\mathbb{P}_t, \mathbb{V}_t, \Delta\mathbb{V}_t, \mathbb{A}_t]$ and $\overrightarrow{\phi}_t$ is the actual vector representation of the predicted ϕ_t in the model $\phi_t \in \{-1, 0, 1\}$, where -1 is left lane change, 0 is no lane change, 1 is right lane change.

Furthermore, when Δt is small enough, the above vehicle longitudinal kinematics also follows the uniform acceleration motion over each Δt. Kinematic equations among the inputs $p_{i,j,t}$, $v_{i,j,t}$, and $a_{i,j,t}$ are as follow:

$$v_{i,j,t} = v_{i,j,t-1} + a_{i,j,t}\Delta t \tag{4}$$

$$p_{i,j,t} = p_{i,j,t-1} + v_{i,j,t}\Delta t + \frac{1}{2}a_{i,j,t}\Delta t^2 \tag{5}$$

By the above mathematical formulation, a specially designed deep neural network is given in the next section.

3. Integrated two-dimensional vehicle trajectory prediction model

In this section, the structure of the integrated two-dimensional vehicle trajectory prediction model is given in detail. Then the key modules and layers are presented.

3.1. Model structure

To build an integrated trajectory prediction model that can accurately predict both lane change and car-following trajectories, a switch structure is developed. The general architecture of the model is shown in Fig. 2. From Fig. 2, the input feed directly into a TCN layer, which is the input feature extractor. The purpose of this TCN layer is to extract temporal domain features. Additionally, this TCN layer can give the model additional depth to the entire model, which can improve prediction performance for deep learning models. Then the extracted features feed into two separated parts of the model, the lane change prediction module, which is represented as the H function in the mathematical formulation

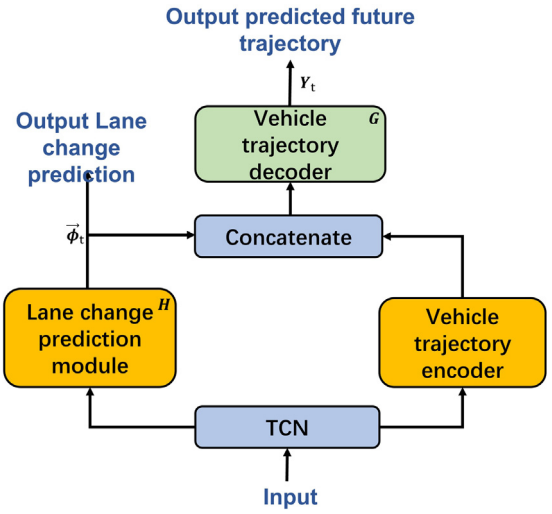


Fig. 2. General model structure of integrated trajectory prediction model.

and the vehicle trajectory encoder. The lane change prediction module will output a lane change prediction output $\overrightarrow{\phi}_t$ that aims to switch between predicting car-following and lane change trajectories, which also creates a unique asymmetric structure. The vehicle trajectory encoder will further refine the extracted features from the TCN layer. Then the output of these modules is concatenated together using matrix concatenation. Finally, the vehicle trajectory decoder, which is represented as the G function in the mathematical formulation, will generate the vehicle trajectory prediction output Y_t based on the concatenated values. The detailed structure of each module and the TCN layer is discussed in the following paragraph.

As shown in Fig. 3, the input historical trajectory data is directly input to TCN layer to extract temporal domain features from the input historical trajectory data. We selected TCN as the main feature extractor, which shows better capability on generalized temporal feature extractor compared with other methods such as BiLSM, as suggested by studies [36,37]. The extracted trajectory features feed into two sections of the model the lane change prediction module and the vehicle trajectory encoder. The lane change prediction module used the extracted features from the TCN layer as inputs and a fully connected layer, a BiLSTM, another fully connected layer, and a softmax layer to generate a lane change prediction $\overrightarrow{\phi}_t$. The output of this is split into two directions. One direction goes into a special fully connect layer to output a lane change prediction. The fully connected layer aims to use the calculated probability from the softmax layer and output a predicted lane change label to indicate whether the vehicle is changing lanes or not. The other direction will concatenate together with the output of the other section of the model. For the other section, the output of the TCN layer feeds directly into the vehicle trajectory encoder built using a BiLSTM layer and an attention layer. Between the BiLSTM and the attention layer, there is a normalization layer to enhance the prediction performance. The normalization layer of this framework is the layer norm method from [38]. This type of normalization layers can be used to stabilize the training process of recurrent neural networks like BiLSTM, which would greatly improve the prediction performance of model. In addition, using a layer normalization layer can also reduce the total training time by improving the training convergence rate. Then, the attention layer's output matrix is concatenated together with the lane change prediction result using matrix concatenation, which will go into the vehicle trajectory decoder to generate the predicted future trajectories Y_t of the current vehicle. The matrix concatenation process is described as follows:

The output from the softmax layer is a feature matrix A, the output of the vehicle trajectory encoder is a feature matrix B, and the feature matrix C is the output of the concatenation. The detailed concatenation process is given as below:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ \vdots & \vdots & \vdots \\ a_{d1} & a_{d2} & a_{d3} \end{bmatrix} \tag{6}$$

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{1n} \\ b_{21} & b_{22} & b_{23} & \dots & b_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{d1} & b_{d2} & b_{d3} & \dots & b_{dn} \end{bmatrix} \tag{7}$$

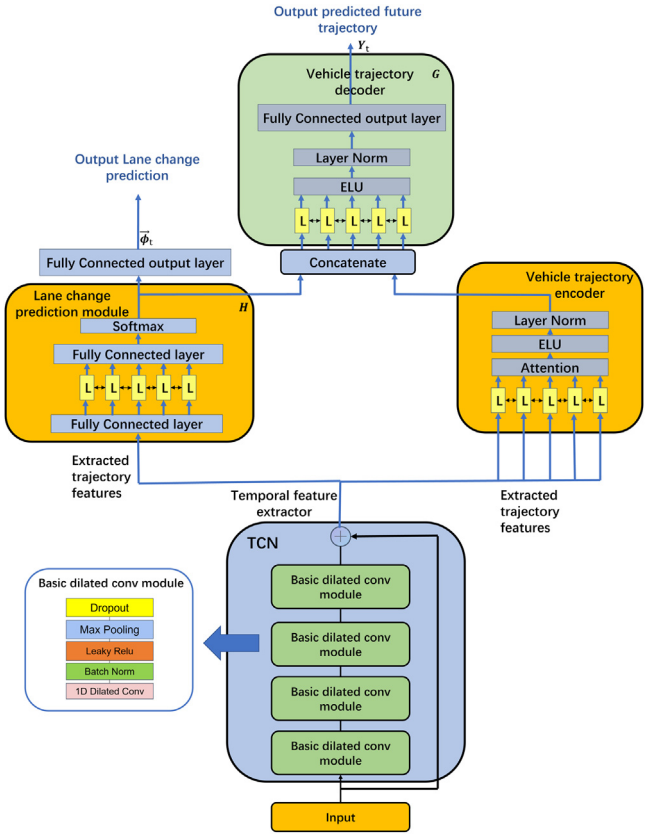


Fig. 3. Detailed model structure of our integrated trajectory prediction model.

$$AconcatB = C = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_{11} & \dots & b_{1n} \\ a_{21} & a_{22} & a_{23} & b_{11} & \dots & b_{2n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{d1} & a_{d2} & a_{d3} & b_{d1} & \dots & b_{dn} \end{bmatrix} \tag{8}$$

The vehicle trajectory decoder is built using a BiLSTM layer, an ELU activation layer, a normalization layer, and a fully connected output layer. We choose ELU activation for the vehicle trajectory encoder and the decoder in the model because the ELU activation function is a better activation function for learning sequence based data and provides better generalization abilities that will be critical for encoding and decoding vehicle trajectory features. The following formula represents the ELU activation function:

$$ELU\ (x) = \exp\ (x) - 1 \tag{9}$$

The parameters and the detailed input and output design of the layers are designed using empirical trials and tuning. The detailed input and output size for each layer in the model is described in Table 1.

3.2. Temporal convolution neural network layer

In order to successfully use convolutions in sequence-based learning tasks, we exploit the TCN model from WaveNet [35], which uses a unique convolution method called dilated casual convolution. For a convolution operation, the receptive field indicates the length of the historical information that can be utilized for the output of the convolution. Thus, a large receptive field can increase the amount of historical information that can be utilized for prediction, which tends to lead to better prediction performance of time series tasks. The dilated convolution use dilation factors to increase the size of the receptive field. Stacking this dilated convolution with a different number of dilation factors will create a special structure that will significantly increase the receptive field of the convolution model. Given a 1-D input sequence:

Table 1
Input–output size for each layers in the integrated two dimensional trajectory prediction model.

Layers	Input size	Output size
TCN	45	20
Lane change prediction module (fully connected layer)	20	40
Lane change prediction module (BiLSTM)	40	40
Lane change prediction module (fully connected layer)	40	3
Lane change prediction module (softmax layer)	3	1
Lane change prediction module (fully connected output layer)	3	1
Vehicle trajectory encoder (BiLSTM)	20	20
Vehicle trajectory encoder (attention layer)	20	20
Vehicle trajectory encoder (ELU layer)	20	20
Vehicle trajectory encoder (layer norm)	20	20
Vehicle trajectory decoder (fully connected layer)	23	80
Vehicle trajectory decoder (BiLSTM)	80	80
Vehicle trajectory decoder (fully connected layer)	80	Output size

$x \in \mathbb{R}^n$, and a kernel w: $\{0, 1, \ldots, k-1\} \rightarrow R$, the 1D dilated causal convolution $(x *_d w)$ is

$$(x *_d w) = \sum_{i=0}^{k-1} w(i)x(s - id), \tag{10}$$

where k and d are the kernel size and the dilation factor respectively. With a large dilation factor d, larger receptive fields can be achieved by stacking multiple layers of TCN.

To properly extract temporal domain features for accurate vehicle trajectory prediction, only relying on dilated convolution layers is not sufficient. Thus, we added additional activation, normalization, pooling, and dropout layers. Also, to further improve the prediction performance, an additional residual connection is added to the layer.

The activation layer adds essential nonlinearity to the TCN model, which will help the TCN model to learn critical nonlinear relationships and patterns from the input data. We used Leaky ReLU as an activation function for our TCN layer because our inputs are the relative positions that are often negative values, which can cause regular ReLU activation problems in training. Leaky ReLU can reduce these problems for negative inputs. The following formula defines the leaky ReLU activation function:

$$LeakeyReLU(x) = \begin{cases} ax & \text{x}<0 \\ x & \text{x}\geq 0 \\ , \end{cases} \tag{11}$$

where a is the slope parameter for the negative section of the Leaky ReLU activation function. a should be a very small value.

The dilated convolution layers will output a feature map from the input. This feature map often contains many features that are not necessarily significant and can cause bias and errors in prediction. Therefore, a downsampling technique is applied to the feature map, which is called pooling. Max pooling is used in our TCN layer because average pooling will generate a feature map that could make distinguishing between car-following and lane changing maneuvers more difficult for the model.

Normalization layers are a common method to improve the prediction performance of deep learning models. One of the most common normalization layers is batch normalization which was developed by Google in 2015 [39]. The batch normalization layer can standardize the input to each deep learning layer for each mini-batch which can stabilize the training process and improve the model prediction performance. Therefore, we implemented batch normalization layers in between the convolution layers inside the TCN layer.

The residual connection comes from one of the most classic CNN based deep learning models called ResNet [40]. Adding this skip connection can preserve the original gradient and help model training so that the gradient does not vanish during the training process of a deep neural network. This method has been proven to improve the training and general performance of deep CNNs. Thus, a residual connection is added to the TCN layer. The following formula defines the skip connection.

$$o = Activation(F(X_t) + X_t). \tag{12}$$

The dropout layer was developed by Srivastava et al. [41] as a simple solution to reduce the overfitting problems of neural networks. Dropout layers are added to the TCN layer to reduce the overfitting problem learned by the model. There is a drop layer for every convolution layer in the TCN layer.

3.3. LSTM

LSTM is an improved version of RNN. LSTM has been used in many time series related fields due to its ability to learn long term dependencies extremely well. For every time step n, the LSTM cell has an input x_n, a output h_n and a cell state

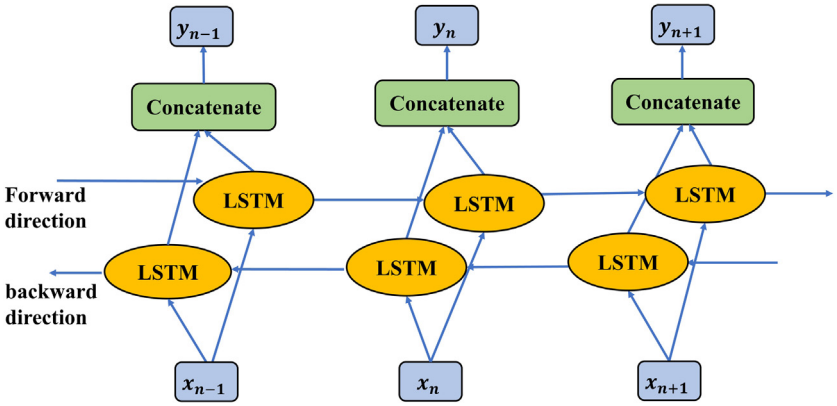


Fig. 4. The general structure of a BiLSTM layer.

output C_n. The LSTM cell can update the output based on the cell state input C_{n-1}, the hidden state input h_{n-1}, and the input x_n. The LSTM cell uses three gates to control the information in the cell state, including the forget gate f_n, the input gate i_n, and the output gate o_n. These three gates give the LSTM cells ability to learn long term dependency. The three gates control what information should pass through the cells and what information should be updated through the layer. The calculation process of the LSTM cell and the outputs are shown in the following equations:

$$\begin{aligned} f_n &= \sigma_g\,(W_f x_n + U_f h_{n-1} + b_f),\\ i_n &= \sigma_g\,(W_i x_n + U_i h_{n-1} + b_i),\\ o_n &= \sigma_g\,(W_o x_n + U_o h_{n-1} + b_o),\\ C_n &= f_n \odot C_{n-1} + i_n \odot \tanh(W_C x_n + U_C h_{n-1} + b_C),\\ h_n &= o_n \odot \tanh(C_n), \end{aligned} \tag{13}$$

where n represents the input to the LSTM cell for the model. W_f, W_i, W_o, and W_C are weight vectors for each gate. U_f, U_i, U_o, and U_C are weight vectors associated with the previous output vector for each gate and cell state output. $\odot$ is a dot product operation between vectors. The *tanh* is the hyperbolic tangent function. σ_g represents the gate activation function, and the sigmoid function is used in our study.

3.4. BiLSTM

Bidirectional LSTM(BiLSTM) is one of the successful variants of improved LSTM [42]. Compared to the traditional LSTM, BiLSTM has the ability to learn relationships from the input from both the forward direction and the backward direction. As shown in Fig. 4, the structure of BiLSTM is created using LSTM cells from two layers, the forward and backward layers. Therein, the output of the forward layer can be calculated using the input information in a positive (forward) direction from time step $t-1$ to t, and the output of the backward layer is obtained using the reversed input information from time step $t+1$ to t. To be noted that the backward and forward layers do not share parameters. The BiLSTM contains two sets of training parameters; one set governs how information flows forward, the other controls how information flows backward. This unique bidirectional structure of BiLSTM can effectively learn both forward and backward dependencies and pass through more useful input information [43]. With this design, BiLSTM can process sequence data and learn temporal domain features using both the forward and backward layers, which indicates that more information can be used for prediction and will lead to improved prediction performance. The output y_n of BiLSTM at n is calculated using the output of the forward LSTM layer $\overrightarrow{h_n}$ and backward LSTM layer output $\overleftarrow{h_n}$, as in the equation below:

$$y_n = [\overrightarrow{h_n} \oplus \overleftarrow{h_n}], \tag{14}$$

where $\oplus$ is a type of combination function that combines $\overrightarrow{h_n}$ and $\overleftarrow{h_n}$. Several functions can be used as $\oplus$, including sum, average, concatenate, and multiplication. In our study, we select the concatenate function as the combination function, because we want all the features extracted by the LSTMs to be available for prediction.

3.5. Attention mechanism

To further improve the prediction performance of the BiLSTM/LSTM model, the attention mechanism is added to the model. Fig. 5 shows our attention mechanism. For a BiLSTM/LSTM model, the final output is often a context vector representing all the input's important features. This would be challenging for a single BiLSTM/LSTM cell to capture all

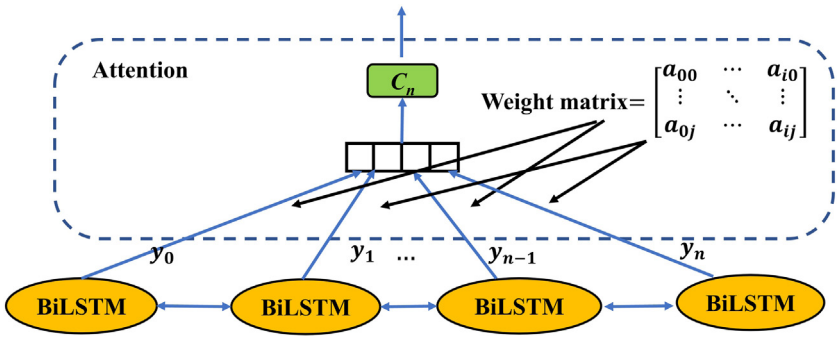


Fig. 5. The general structure of our attention layer.

the required temporal information when the input sequence is very long. The attention mechanism helps the model capture temporal features from a long input sequence by generating the output vectors from all BiLSTM/LSTM cells in a layer instead of relying on the last cell. The attention mechanism also selects the most critical output features from those BiLSTM/LSTM output using learned weights based on the importance of these features learned from the training data using a feed-forward network. The output result of the attention layers is a weighted sum of the BiLSTM/LSTM cells' output using the learned weights. Such an attention layer can highlight the importance of different outputs, potentially represent a particular feature, and consider more contextual impacts to improve prediction performance. The following functions are used to show the calculation process of the attention mechanism used in our study:

$$\begin{aligned} c_n &= \sum_{i=0}^{N} a_{n,i} \cdot y_n, \\ a_{n,i} &= \frac{\exp(\beta(y_i))}{\sum_{k=0}^{N} \exp(\beta(y_k))}, \\ \beta(y_i) &= {V_a}^T \tanh(W_a y_i), \end{aligned} \tag{15}$$

where c_n is the output vector. N is the total length of the input data. y_n is the output of the last BiLSTM layer at position n. $a_{n,i}$ is the alignment model that assigns a weight at each location i for the BiLSTM output at position i. The alignment score is calculated using the β function. The β function is a score function that is learned from a particular feed-forward network that only contains one hidden layer and is trained together with all the other parts of the model. V_a and W_a are learned weights from the alignment model. *tanh* is the hyperbolic tangent function.

3.6. Model training

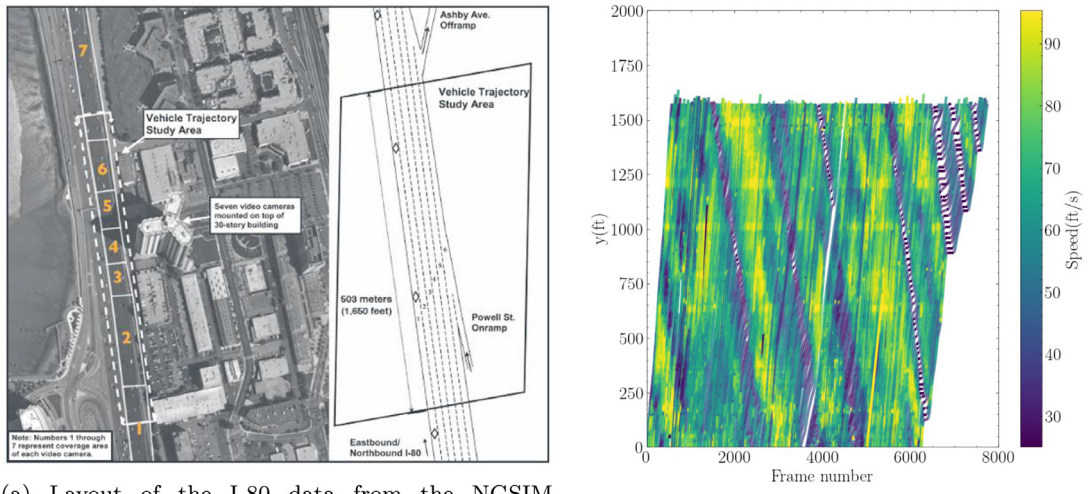
As our model contains two parts, the training loss for our model also contains two parts. The first part is the mean square error(MSE) loss which is used for the trajectory prediction output. The second part is a categorical cross-entropy loss. The following formula shows how our combined loss is calculated:

$$L = \frac{1}{n}\sum_{i=0}^{n}(Y_i - \hat{Y}_i)^2 - \sum_{k=0}^{2} y_k log(\hat{y_k}), \tag{16}$$

where Y_i is the actual position of the vehicle and $\hat{Y}_i$ is the predicted position. y_k is the actual movement type of the vehicle and $\hat{y_k}$ is the predicted movement type. n is the prediction horizon and maximum value for k is 2 because we have 3 movement types.

4. Experiments and results

In this section, the training data, the evaluation metrics, the experiment design, and the experiment results are presented in details. The experiments are conducted using a computer with RTX 3090 GPU and a Ryzen 3700x CPU and the models are developed by python and the Pytorch deep learning framework. The experiment contains three sections. The first section gives the model training process which determines the hyper-parameters . The second experiment conducts a sensitivity analysis under different prediction/memory horizons. For the third experiment, we make a systematic comparison with state of art vehicle trajectory prediction methods.



(a) Layout of the I-80 data from the NGSIM dataset[44]

(b) Sample trajectory from the NGSIM dataset

Fig. 6. Sample trajectory and layout of the NGSIM dataset.

4.1. Data description

Data-driven based vehicle trajectory prediction models rely on a large amount of training data. Thus, our integrated two dimensional trajectory prediction model requires many real world vehicle trajectory data to train and test the model properly. We used one of the most popular public available real world vehicle trajectory dataset, the Next Generation Simulation (NGSIM) dataset [44] to train and test our model. The data used in the experiment is collected from US Highway 101 (US-101) and Interstate 80 Freeway (I-80), in a 10 Hz frequency. The NGSIM dataset contains trajectory data recorded from over 3000 vehicles. 2250 trajectories are used to train the model, and 250 are used to test the model. The detailed layout of the dataset is shown in Fig. 6(a) and sample trajectories are shown in Fig. 6(b).

4.2. Quantitative evaluation metrics

To quantitatively evaluate the prediction performance of our model, the following Mean absolute error (MAE), Root-mean-square error (RMSE) are selected as car following trajectories evaluation metrics as below:

$$MAE = \frac{\sum_{i=0}^{K} |Y_i - \widehat{Y}_i|}{K}, \tag{17}$$

$$RMSE = \sqrt{\frac{\sum_{i=0}^{K} (Y_i - \widehat{Y}_i)^2}{K}}, \tag{18}$$

Where Y_i is the groundtruth position of the vehicle; $\widehat{Y}_i$ is the predicted output position from the model and K is the total number of predictions.

Further, we also evaluated the lane change prediction accuracy by using F1 score, whose detailed definitions and equations are given as below:

Accuracy: The accuracy is calculated by using the amount of true positive and true negative predicted by the model divided by the total number of test samples.

Precision: Precision is defined as the number of true positives predicted by the model divided by the number of test samples. For each of the labels in the data, an individual precision score needs to be calculated.

Recall: Recall is defined as the number of true positives predicted by the model divided by the number of true positives plus false negatives predicted by the model. An individual recall score is also calculated for each of the labels in the data.

F1 score: F1 score is a ratio created using precision and recall. It is used to show the balance of precision and recall for a model, because sometimes high precision or high recall does not translate to good prediction performance. The F1 score is defined from the following formula:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}. \tag{19}$$

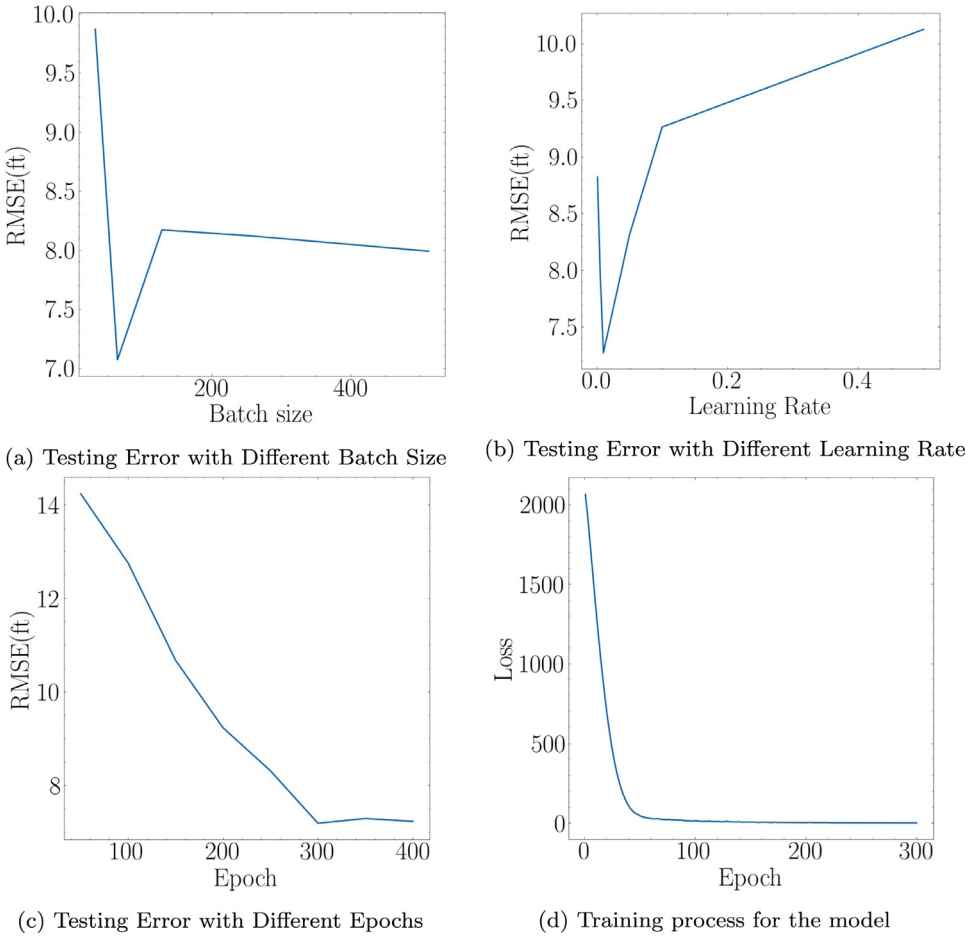


Fig. 7. Model training experiment result.

Table 2
Model parameters for the integrated two dimensional trajectory prediction model.

Parameters	Values
Learning rate	0.01
Learning rate decay factor	0.0005
Epoch	300
Batch size	64
Optimizer	Adam

4.3. Model training

In this model training experiment, the model hyper-parameters, including learning rate, training epoch, and batch size, are determined by trying different values. Specifically, we use RMSE as a metric to select the best hyper-parameters. The details of training are as illustrated by Fig. 7. Fig. 7(a) shows how the model testing RMSE changes with different batch sizes. The effect of different learning rates and Epochs on the RMSE is shown in Fig. 7(b) and Fig. 7(c), respectively. The selected hyperparameters from the experiment are shown in Table 2. As we can find from Fig. 7(d), the training process is smooth. The training loss decreases significantly for the first 50 epochs. Then the loss decrease begins to slow down and finally converges at the end of the training process. This result manifests that the model is trained in a good manner. The total training time for this model is around 35 min.

Table 3
Model general performance experiment result.

Model	MAE	RMSE	Accuracy	F1
ITPM	**2.11**	**5.67**	**0.8789**	**0.8372**
ITPM without TCN layer	2.79	6.23	0.8220	0.8011
ITPM without attention layer	2.63	6.01	0.8720	0.8241
ITPM BiLSTM feature extractor	2.59	6.14	0.8512	0.8193
Baseline BiLSTM [46]	3.67	8.36	N/A	N/A
Baseline LSTM [45]	4.21	9.37	N/A	N/A
Baseline IDM+MOBIL [14,19]	6.23	12.72	0.64879	0.6331
ConvLSTM [47]	3.21	6.77	N/A	N/A

Table 4
Model prediction efficiency experiment result.

Model	Parameters	Inference time
ITPM	861084	21.7 ms
ITPM without TCN layer	770784	17.8 ms
ITPM BiLSTM feature extractor	832198	22.8 ms
ITPM without attention layer	843751	19.5 ms
Baseline BiLSTM [46]	323756	8.7 ms
Baseline LSTM [45]	157480	3.7 ms
Baseline IDM+MOBIL [14,19]	N/A	N/A
ConvLSTM [47]	467338	14.6 ms

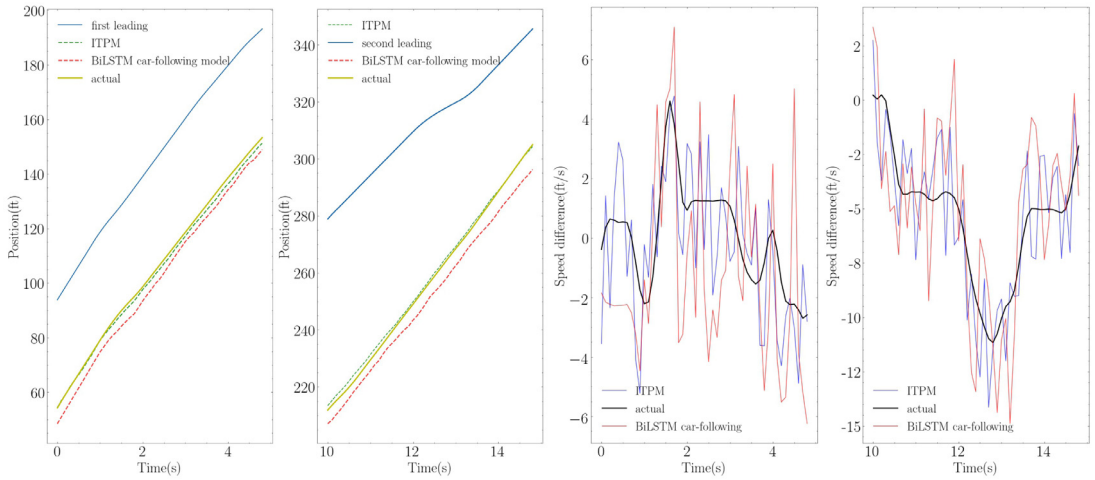
4.4. Model prediction performance comparison

In this section, experiments are conducted to compare the prediction performance of our integrated two dimensional trajectory prediction model with several baseline methods to validate the effectiveness of our model. We selected the MOBIL [19] model as a baseline method because it is a classical lane change model representing the model-based method discussed in the introduction section. The LSTM [45] and the BiLSTM [46] models are baselines representing the data-driven based methods. Additionally, we added the ConvLSTM [47] method to compare our model with an advanced data-driven method. The integrated trajectory prediction model and the baseline models are quantitatively evaluated using the evaluation metrics discussed in the previous section. The memory horizon and prediction horizon for this experiment is 5s for both models. The quantitative experiment result for this experiment is shown in Table 3.

First, the MAE of the integrated trajectory prediction model is at 2.11 ft, and the RMSE is 5.67 ft. To test the usefulness of TCN layer, we conduct a comparison with the model without TCN layer, whose MAE and RSME are 2.79 ft and 6.23 ft, respectively, which suggests the usefulness of TCN. In addition, we test a case which applies BiLSTM as the feature extractor, whose MAE is 2.59ft and RMSE is 6.14ft. This result shows the advantage of using TCN as the feature extractor. By comparing the result with the model without the attention layer, whose MAE and RMSE are 2.63ft and 6.01ft, we can find the effectiveness of the attention layer. Further, the BiLSTM model gives MAE and RMSE as 3.67 ft and 8.36 ft, which is significantly larger than that of our method. This validates the advantages of our switch structure. The MAE and RMSE for the LSTM model are 4.21ft and 9.37ft. Therefore, by comparing BiLSTM with LSTM, we can find that the effectiveness of the bidirectional structure on capturing the trajectory time series feature. Moreover, traditional model-based prediction method, IDM+MOBIL, whose MAE and RMSE are 6.23 ft and 12.72 ft, suggests the weakness of traditional model-based prediction accuracy. In addition, convLSTM is a relatively advanced data driven method, and its MAE and RMSE are 3.21 ft and 6.77ft, which demonstrate the effectiveness of our model comparatively. For the lane change prediction performance, our model achieved a better prediction accuracy, 0.8789, compared with that of the IDM+MOBIL model, which is 0.6879. The F1 score also gives a similar conclusion.

The prediction efficiency of the models is of interest in our study. To compare it, we recorded the parameters required and inference time of each model, which is shown in Table 4. From the result, it is clear that our model uses 861084 parameters and 21.7 ms inference time. Compared with the LSTM and BiLSTM models, our model uses more parameters and inference time as expected by the additional complexity of our switch structure. Whereas, the prediction efficiency of our model is still adequate for automated driving and other intelligent system transportation applications since the inference time is significantly less than 30 ms.

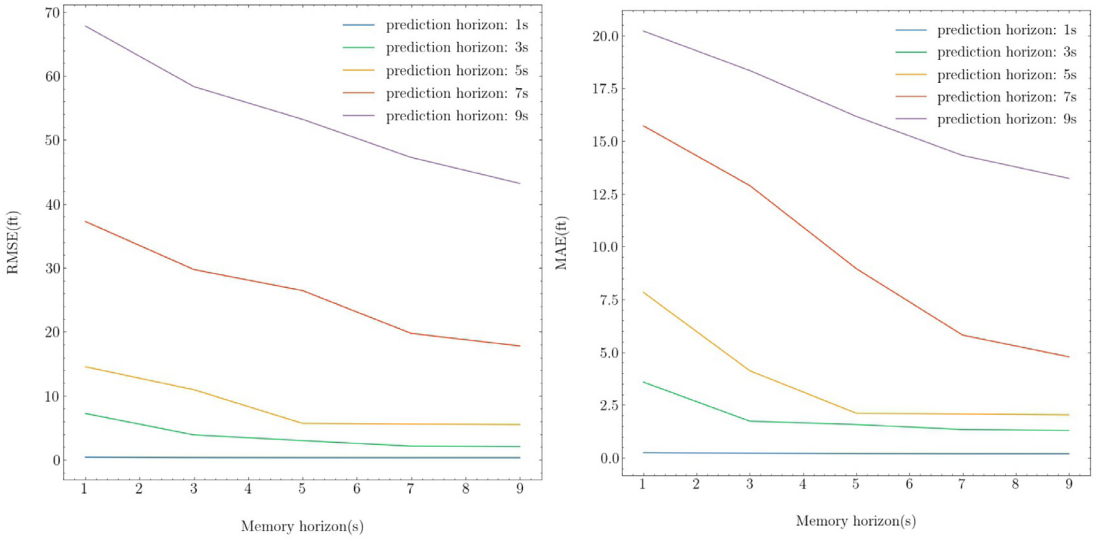
To qualitatively compare the BiLSTM car-following model with our model. A special case before and after lane change is selected as an illustration. The memory horizon and prediction horizon for this experiment is 3s for both models. The speed difference and position from this experiment are shown in Figs. 8(a) and 8(b). As indicated by Figs. 8(a) and 8(b), our model performs better than the car-following model in both position and speed difference accuracy. As shown in Fig. 8(a), the green line, which is the predicted position from our model, is closer to the actual position of the vehicle compared to the BiLSTM model, which is shown as the red line in the figure, for both before the lane change and after the lane change. It is clear that from Fig. 8(b), the predicted speed difference is which is the blue line is closer to the black line, which is the actual speed difference compared to the red line, which is the predicted speed difference from



(a) Comparison with BiLSTM car-following model experiment position plot

(b) Comparison with BiLSTM car-following model experiment speed difference plot

Fig. 8. Comparison with BiLSTM car-following model experiment result.



(a) The RMSE error of the integrated trajectory prediction model trained with different prediction/memory horizon

(b) The MAE error of the integrated trajectory prediction model trained with prediction/memory horizon

Fig. 9. Model performance with different prediction/memory horizon.

the BiLSTM model for both sections of the experiment. We can also find that our model produces relatively smoother predictions than BiLSTM in terms of the speed difference.

4.5. Model performance sensitivity analysis

To further investigate the performance of our proposed model, we conduct a sensitivity analysis with different memory horizons m and prediction horizons n. The detailed RMSE and MAE are given in Figs. 9(a) and 9(b). As shown in Figs. 9(a) and 9(b), the MAE and RMSE errors for the 1s prediction/memory horizon are both less than 1ft, and the MAE and RMSE for the 3s prediction/memory horizon are both less than 10ft, which suggests the prediction excellence for the short-term prediction under different memory horizon. Further, from the short-term prediction, we can find that, as long as the memory horizon is larger than or equals to 3 s, the performance is relatively stable, while if the memory horizon is 1 s, the prediction accuracy plunged during lack of necessary information. Moreover, the experiment result shows

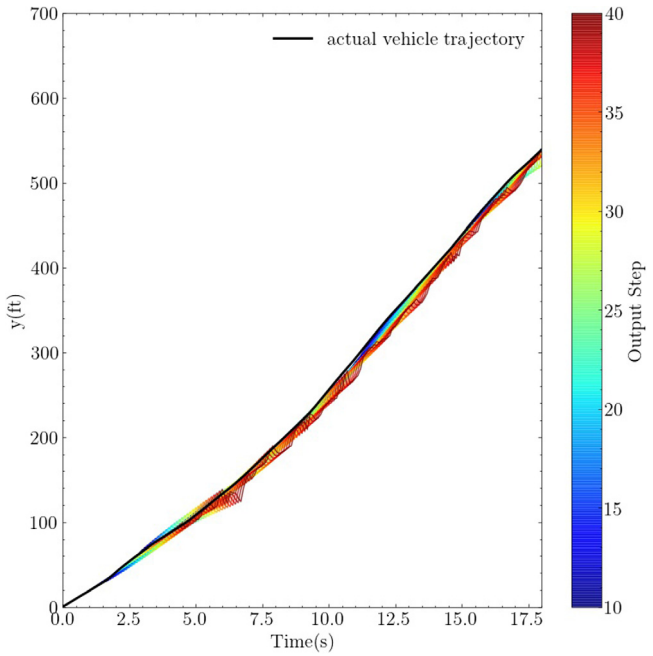


Fig. 10. Example trajectory predicted by the integrated trajectory prediction model.

that the prediction performance started to decrease more significantly when the prediction/memory horizon increased to larger than 5s, due to the error propagation, which is a natural phenomenon. We can also find that, in order to achieve better performance for long-term prediction, a larger memory horizon is more favored. As an illustration, an example of trajectory prediction with different prediction horizons given $m = 5$ s is shown in Fig. 10. In Fig. 10, the memory horizon is fixed as 5 s, and we plot the trajectories predicted by our model with different prediction horizons to demonstrate the ability of our model for giving both shorter and long-term accurate predictions. In conclusion, the results from this experiment show that our integrated trajectory prediction model can solve both short term prediction and long term prediction. This means that our model can effectively learn both features for short term prediction and long term prediction.

5. Conclusion

In this paper, an integrated two dimensional trajectory prediction model is proposed to capture the joint behaviors of car following and lane change. Differed from the state of art approaches, which largely treat these two motions as independent processes, our model is specifically designed in an integration framework to model the dependencies. We firstly systematically formulate the above mentioned problem, based on which a special designed deep neural network with a switch structure utilizing the TCN layer, BiLSTM, and the attention mechanism is constructed. The switch component aims to determine the prediction mode with/without lane change.

Experiments are conducted to evaluate and validate our proposed model and the effectiveness of the specially designed deep neural network. Firstly, we conducted a model performance experiment by comparing our model with the baseline methods using both quantitative and qualitative evaluation methods. The experimental results show that the proposed integrated two dimensional trajectory prediction model has a better prediction performance compared with the state of art models. The comparison results also validate the effectiveness of each component in our neural network structure. Secondly, we have conducted a prediction/memory horizon sensitivity analysis to give practitioners guidance for the choice of these parameters. Further, we can find from the result that our model can make effective short term and long term trajectory predictions.

For future work, there are several aspects that can be improved based on this research. Firstly, models for platoon level trajectory prediction [48] considering the inter-connection between the vehicles in the platoon can be further developed. Secondly, we will further incorporate the heterogeneity of human driving behaviors [49] in our model.Thirdly, we will try to investigate the possibility of leveraging insights from fluid dynamics [50,51] to trajectory prediction.

CRediT authorship contribution statement

Kunsong Shi: Conceptualization, Methodology, Data collection, Formal analysis, Software, Writing – original draft, Writing – review & editing. **Yuankai Wu:** Conceptualization, Methodology, Formal analysis, Software, Writing – original

draft, Writing – review & editing. **Haotian Shi:** Data collection, Formal analysis, Software, Writing – original draft, Writing – review & editing. **Yang Zhou:** Conceptualization, Methodology, Formal analysis, Software, Writing – original draft, Writing – review & editing. **Bin Ran:** Conceptualization, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research is funded by the University of Wisconsin Traffic Operation and Safety Laboratory.

References

[1] N. Deo, M.M. Trivedi, Convolutional social pooling for vehicle trajectory prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 1468–1476.
[2] Y. Zhou, S. Ahn, M. Chitturi, D.A. Noyce, Rolling horizon stochastic optimal control strategy for ACC and CACC under uncertainty, Transp. Res. C 83 (2017) 61–76.
[3] X. Wu, H. Chen, C. Chen, M. Zhong, S. Xie, Y. Guo, H. Fujita, The autonomous navigation and obstacle avoidance for USVs with ANOA deep reinforcement learning method, Knowl.-Based Syst. (ISSN: 0950-7051) 196 (2020) 105201, http://dx.doi.org/10.1016/j.knosys.2019.105201, URL https://www.sciencedirect.com/science/article/pii/S0950705119305350.
[4] W. Gao, Z.-P. Jiang, K. Ozbay, Data-driven adaptive optimal control of connected vehicles, IEEE Trans. Intell. Transp. Syst. 18 (5) (2016) 1122–1133.
[5] X. Liu, D. Shen, L. Lai, S. Le Vine, Optimizing the safety-efficiency balancing of automated vehicle car-following, Accid. Anal. Prev. 136 (2020) 105435.
[6] H. Chae, M. Lee, K. Yi, Probabilistic prediction based automated driving motion planning algorithm for lane change, in: 2017 17th International Conference on Control, Automation and Systems, ICCAS, IEEE, 2017, pp. 1640–1645.
[7] H. Shi, Y. Zhou, K. Wu, X. Wang, Y. Lin, B. Ran, Connected automated vehicle cooperative control with a deep reinforcement learning approach in a mixed traffic environment, Transp. Res. C 133 (2021) 103421.
[8] H. Shi, Y. Zhou, X. Wang, S. Fu, S. Gong, B. Ran, A deep reinforcement learning-based distributed connected automated vehicle control under communication failure, Comput.-Aided Civ. Infrastruct. Eng. (2022).
[9] H. Shi, Q. Nie, S. Fu, X. Wang, Y. Zhou, B. Ran, A distributed deep reinforcement learning–based integrated dynamic bus control system in a connected environment, Comput.-Aided Civ. Infrastruct. Eng. (2021).
[10] Y. Hou, P. Edara, C. Sun, A genetic fuzzy system for modeling mandatory lane changing, in: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, IEEE, ISBN: 9781467330640, 2012, pp. 1044–1048, http://dx.doi.org/10.1109/ITSC.2012.6338877.
[11] J. Nie, J. Zhang, X. Wan, W. Ding, B. Ran, Modeling of decision-making behavior for discretionary lane-changing execution, in: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, IEEE, ISBN: 9781509018895, 2016, pp. 707–712, http://dx.doi.org/10.1109/ITSC.2016.7795631.
[12] L.A. Pipes, An operational analysis of traffic dynamics, J. Appl. Phys. 24 (3) (1953) 274–281, http://dx.doi.org/10.1063/1.1721265.
[13] H. Gong, H. Liu, B.-H. Wang, An asymmetric full velocity difference car-following model, Physica A 387 (11) (2008) 2595–2602.
[14] M. Treiber, A. Hennecke, D. Helbing, Congested traffic states in empirical observations and microscopic simulations, Phys. Rev. E 62 (2) (2000) 1805.
[15] X. Li, X. Wang, Y. Ouyang, Prediction and field validation of traffic oscillation propagation under nonlinear car-following laws, Transp. Res. B 46 (3) (2012) 409–423.
[16] D. Chen, J. Laval, Z. Zheng, S. Ahn, A behavioral car-following model that captures traffic oscillations, Transp. Res. B 46 (6) (2012) 744–761.
[17] P.G. Gipps, A model for the structure of lane-changing decisions, Transp. Res. B (ISSN: 01912615) 20 (5) (1986) 403–414, http://dx.doi.org/10.1016/0191-2615(86)90012-3.
[18] P. Hidas, Modelling vehicle interactions in microscopic simulation of merging and weaving, Transp. Res. C (ISSN: 0968090X) 13 (1) (2005) 37–62, http://dx.doi.org/10.1016/j.trc.2004.12.003.
[19] A. Kesting, M. Treiber, D. Helbing, General lane-changing model MOBIL for car-following models, Transp. Res. Rec. 1999 (1) (2007) 86–94.
[20] J.A. Laval, L. Leclercq, Microscopic modeling of the relaxation phenomenon using a macroscopic lane-changing model, Transp. Res. B (ISSN: 01912615) 42 (6) (2008) 511–522, http://dx.doi.org/10.1016/j.trb.2007.10.004.
[21] L. Zhao, Z. Li, A.Y. Al-Dubai, G. Min, J. Li, A. Hawbani, A.Y. Zomaya, A novel prediction-based temporal graph routing algorithm for software-defined vehicular networks, IEEE Trans. Intell. Transp. Syst. (2021) 1–16, http://dx.doi.org/10.1109/TITS.2021.3123276.
[22] S. Patel, B. Griffin, K. Kusano, J.J. Corso, Predicting future lane changes of other highway vehicles using RNN-based deep models, 2018, URL http://arxiv.org/abs/1801.04340.
[23] Z. Cui, R. Ke, Y. Wang, Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction, 2018, pp. 1–12, URL http://arxiv.org/abs/1801.02143.
[24] M. Zhou, X. Qu, X. Li, A recurrent neural network based microscopic car following model to predict traffic oscillation, Transp. Res. C 84 (2017) 245–264.
[25] Z. Zhang, F. Ding, Y. Zhou, S. Ahn, B. Ran, Deep Long Short-Term Memory Network Based Long-Term Vehicle Trajectory Prediction, Tech. rep., 2019.
[26] L. Zhao, Y. Liu, A.Y. Al-Dubai, A.Y. Zomaya, G. Min, A. Hawbani, A novel generation-adversarial-network-based vehicle trajectory prediction method for intelligent vehicular networks, IEEE Internet Things J. 8 (3) (2021) 2066–2077, http://dx.doi.org/10.1109/JIOT.2020.3021141.
[27] L. Ma, S. Qu, A sequence to sequence learning based car-following model for multi-step predictions considering reaction delay, Transp. Res. C 120 (2020) 102785.
[28] D.F. Xie, Z.Z. Fang, B. Jia, Z. He, A data-driven lane-changing model based on deep learning, Transp. Res. C (ISSN: 0968090X) 106 (October 2018) (2019) 41–60, http://dx.doi.org/10.1016/j.trc.2019.07.002.
[29] O. Scheel, N.S. Nagaraja, L. Schwarz, N. Navab, F. Tombari, Attention-based lane change prediction, in: Proceedings - IEEE International Conference on Robotics and Automation, vol. 2019, ISBN: 9781538660263, 2019, pp. 8655–8661, http://dx.doi.org/10.1109/ICRA.2019.8793648, arXiv:1903.01246.

[30] J. Gao, Y.L. Murphey, H. Zhu, Multivariate time series prediction of lane changing behavior using deep neural network, Appl. Intell. (ISSN: 15737497) 48 (10) (2018) 3523–3537, http://dx.doi.org/10.1007/s10489-018-1163-9.
[31] D. Ngoduy, Effect of the car-following combinations on the instability of heterogeneous traffic flow, Transp. B 3 (1) (2015) 44–58.
[32] S. Ahn, M.J. Cassidy, Freeway traffic oscillations and vehicle lane-change maneuvers, in: Transportation and Traffic Theory 2007. Papers Selected for Presentation At ISTTT17Engineering and Physical Sciences Research Council (Great Britain) Rees Jeffreys Road FundTransport Research FoundationTMS ConsultancyOve Arup and Partners, Hong KongTransportation Planning (International) PTV AG, 2007.
[33] F. Marczak, W. Daamen, C. Buisson, Empirical analysis of lane changing behavior at a freeway weaving section, in: 93rd Annual Meeting of the Transportation Research Board, Washington, DC, 2014.
[34] Z. Zheng, S. Ahn, D. Chen, J. Laval, The effects of lane-changing on the immediate follower: Anticipation, relaxation, and change in driver characteristics, Transp. Res. C (ISSN: 0968-090X) 26 (2013) 367–379, http://dx.doi.org/10.1016/j.trc.2012.10.007, URL https://www.sciencedirect.com/science/article/pii/S0968090X12001295.
[35] A.v.d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, Wavenet: A generative model for raw audio, 2016, arXiv preprint arXiv:1609.03499.
[36] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018, arXiv preprint arXiv:1803.01271.
[37] R. Wan, S. Mei, J. Wang, M. Liu, F. Yang, Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting, Electronics 8 (8) (2019) 876.
[38] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, 2016, arXiv:1607.06450.
[39] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015, CoRR abs/1502.03167 arXiv:1502.03167.
[40] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (56) (2014) 1929–1958, URL http://jmlr.org/papers/v15/srivastava14a.html.
[42] S. Siami-Namini, N. Tavakoli, A.S. Namin, The performance of LSTM and BiLSTM in forecasting time series, in: Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019, IEEE, ISBN: 9781728108582, 2019, pp. 3285–3292, http://dx.doi.org/10.1109/BigData47090.2019.9005997.
[43] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, 2015, CoRR abs/1508.01991 arXiv:1508.01991.
[44] US Department of Transportation, NGSIM - Next generation simulation, 2006.
[45] F. Altché, A. de La Fortelle, An LSTM network for highway trajectory prediction, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems, ITSC, 2017, pp. 353–359, http://dx.doi.org/10.1109/ITSC.2017.8317913.
[46] M. Abdalla, A. Hendawi, H.M.O. Mokhtar, N. Elgamal, J. Krumm, M. Ali, Deep motion: A deep learning system for path prediction using similar motions, IEEE Access 8 (2020) 23881–23894, http://dx.doi.org/10.1109/ACCESS.2020.2966982.
[47] G. Xie, A. Shangguan, R. Fei, W. Ji, W. Ma, X. Hei, Motion trajectory prediction based on a CNN-LSTM sequential model, Sci. China Inf. Sci. 63 (11) (2020) 1–21.
[48] Y. Lin, P. Wang, Y. Zhou, F. Ding, C. Wang, H. Tan, Platoon trajectories generation: A unidirectional interconnected LSTM-based car-following model, IEEE Trans. Intell. Transp. Syst. (2020).
[49] D. Chen, S. Ahn, Capacity-drop at extended bottlenecks: Merge, diverge, and weave, Transp. Res. B (ISSN: 0191-2615) 108 (2018) 1–20, http://dx.doi.org/10.1016/j.trb.2017.12.006, URL https://www.sciencedirect.com/science/article/pii/S0191261517306938.
[50] R. Walters, J. Li, R. Yu, Trajectory prediction using equivariant continuous convolution, in: International Conference on Learning Representations, 2021.
[51] M. Lopez-Martin, S. Le Clainche, B. Carro, Model-free short-term fluid dynamics estimator with a deep 3D-convolutional neural network, Expert Syst. Appl. (ISSN: 0957-4174) 177 (2021) 114924, http://dx.doi.org/10.1016/j.eswa.2021.114924, URL https://www.sciencedirect.com/science/article/pii/S0957417421003651.