

**XUCHUAN LI**, M.Sc.<sup>1,2</sup>  
E-mail: Rayne\_Sun@outlook.com  
**LINKUN FAN**, Master<sup>1,2</sup>  
E-mail: shenywl.think@outlook.com  
**TAO CHEN**, Ph.D.<sup>1,2</sup>  
(Corresponding author)  
E-mail: Chentao@chd.edu.cn  
**CONGSHUAI GUO**, M.Sc.<sup>1,2</sup>  
E-mail: 2578729340@qq.com

<sup>1</sup> Key Laboratory of Automobile Transportation Safety  
Techniques of Ministry of Transport, Chang'an University  
No. 126, Middle of the South Second Ring Road,  
Xi'an, China

<sup>2</sup> School of Automobile, Chang'an University  
No. 126, Middle of the South Second Ring Road,  
Xi'an, China

Traffic Engineering  
Original Scientific Paper  
Submitted: 29 Sep. 2020  
Accepted: 12 Feb. 2021  
DOI: 10.7307/ptt.v33i5.3718

# VEHICLE LANE-CHANGES TRAJECTORY PREDICTION MODEL CONSIDERING EXTERNAL PARAMETERS

## ABSTRACT

*The ability to predict the motion of vehicles is essential for autonomous vehicles. Aiming at the problem that existing models cannot make full use of the external parameters including the outline of vehicles and the lane, we proposed a model to use the external parameters thoroughly when predicting the trajectory in the straight-line and non-free flow state. Meanwhile, dynamic sensitive area is proposed to filter out inconsequential surrounding vehicles. The historical trajectory of the vehicles and their external parameters are used as inputs. A shared Long Short-Term Memory (LSTM) cell is proposed to encode the explicit states obtained by mapping historical trajectory and external parameters. The hidden states of vehicles obtained from the last step are used to extract latent driving intent. Then, a convolution layer is designed to fuse hidden states to feed into the next prediction circle and a decoder is used to decode the hidden states of the vehicles to predict trajectory. The experiment result shows that the dynamic sensitive area can shorten the training time to 75.86% of the state-of-the-art work. Compared with other models, the accuracy of our model is improved by 23.7%. Meanwhile, the model's ability of anti-interference of external parameters is also improved.*

## KEYWORDS

*trajectory prediction; external parameters; dynamic sensitive area; long-short term memory.*

## 1. INTRODUCTION

With the rapid development of intelligent transportation, more and more autonomous vehicles appear on the road. To ensure safety and efficiency of the autonomous driving in the increasingly complex traffic, autonomous vehicles must have the ability to predict the motion of surrounding vehicles. For example, through detecting the insertion of the side vehicle and sudden braking of the vehicle in front, the vehicle has the ability to take reasonable measures to avoid crash. Predicting the trajectory of vehicle accurately is a challenging task. In most cases, the traffic flow is complex and flexible, and vehicles will follow and change lanes for different purposes, which makes the vehicle trajectory highly non-linear [1].

Historical trajectory is used to explore the movement pattern and calculate the probability of next location for historical trajectory. Based on the model-based trajectory prediction [2], traditional prediction models generally combine vehicle dynamics and kinematic model for trajectory prediction [3-6]. For examples, Barth [3] used Kalman filtering and Monte Carlo method to achieve high accuracy in the short-distance uncertainty prediction of the vehicle model. Houenou [6] proposed a method based on a combination of acceleration motion model and polynomial trajectory planning to make up for

shortcomings of model-based methods in long-term prediction. Markov-based models also used similar methods [7, 8]. The abovementioned methods ignore the interactive effects of vehicles and could not make full use of historical information. Therefore, their accuracy declined greatly with the passage of time. In recent years, the research on data-driven trajectory prediction methods became more and more popular [9-11], and the aim is mostly to improve recognition of the lane-changes intention. Many scholars are committed to using recurrent neural networks (RNN) for prediction [12-14]. Liu [15] used hidden Markov models and support vector machine to recognise driving intentions. The most representative methods are the trajectory prediction algorithms based on long-short term memory (LSTM) [16]. Alahi [17] proposed social pooling layer that connected the predicted target and its surrounding targets, which considers the influence of surrounding vehicles. Xu [1] proposed a pooling layer to substitute a convolutional layer on the basis of Social LSTM to make further adjustments to sensitive areas. Liu [18] introduced the attention mechanism on the basis of Social LSTM, which further improved the accuracy of the model to predict long sequences.

However, there are some critical questions in the above data-driven algorithms. Firstly, according to statistics [19], the different vehicles in diverse lanes show large differences in trajectory. However, the input parameters of the current algorithms are confined to the historical trajectory of the vehicle, ignoring the influence of the vehicle's fixed parameters. Secondly, these algorithms define the target area as a circle, which is used to filter out irrelevant vehicles, ignoring the driving characteristics of the vehicle [20]: the vehicles rarely make large movements while driving, and their lateral speed is relatively small on the straight road. In the longitudinal direction, the target vehicle will be affected by surrounding vehicles. Furthermore, the shape and size of the area are directly related to the vehicle speed.

To solve the above problems, aiming at the non-free flow lane-changing behaviour on the straight road, in this paper we propose a dynamic sensitive area based on the relationship between the safe following distance of the vehicle and their speed. Otherwise, based on LSTM, a model that can make full use of the vehicle external parameters is proposed. The main research contents are as follows:

- 1) Dynamic sensitive area: The sensitive area should be the smallest area that includes all the vehicles that may affect the target vehicle. Based on the actual driving data and the safe following distance, a straight line is obtained by fitting with the length of the sensitive area and the speed of the vehicle. Finally, a method for determining the area of interest based on the speed of the vehicle is proposed.
- 2) Predicting trajectory with external parameters of vehicles: A vehicle trajectory prediction model that considers external parameters is proposed. In this paper, vehicle external parameters including the type of vehicle, length and width of vehicles, and lane information are added to the database, so that the model can relate the influence of external parameters while predicting trajectory. In addition, some datasets containing trucks and other types of vehicles are specially selected for training and evaluation while setting dataset.
- 3) Shared LSTM cell: The model proposed by Liu [18] sets up different LSTM cells for the target car and surrounding cars. One of them is used to encode the explicit states, another one is only used as splicing information, which makes it hard to train LSTMs well. In this paper, an LSTM cell is shared by all of the vehicles to speed up the training process and to reduce the model scale.

## 2. DYNAMIC SENSITIVE AREA

The sensitive area should be the smallest area that includes all surrounding vehicles having an effect on the target vehicle. The smaller this area is, the less required calculations will be, and the better real-time performance this model will have. According to the general characteristics of driving [20], the laterally sensitive area of the target vehicle should keep within the width of an adjacent vehicle. According to the national standard GB-50220 [21], the sensitive height of the area is set as 7.3 m in this paper. For the width of the sensitive area, the driver always keeps safe following distance with ahead vehicle in normal driving. Therefore, we propose that the width of the sensitive area can be replaced by the safe following distance, and the speed of the vehicle is used to delimit the width of the rectangular sensitive area (*Figure 1*).

Next Generation Simulation (NGSIM) [22] is a real-world dataset that consists of detailed vehicle trajectory, wide-area detector, and data for researching driver behaviour. In order to get realistic relationship between safe following distance and speed, we

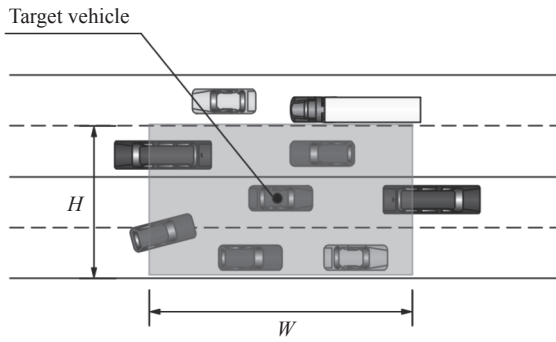


Figure 1 – The proposed sensitive area

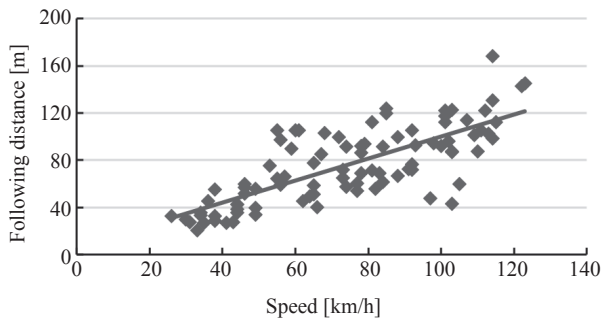


Figure 2 – The relationship between speed and distance

analysed more than 300 car-following data in NG-SIM, and the conducted linear regression is shown in Figure 2.

According to the relationship in Figure 2, an empirical formula is proposed as follows:

$$W = 2 \cdot \begin{cases} 30, & v < 40 \text{ km/h} \\ 0.8v + 20, & v \geq 40 \text{ km/h} \end{cases} \quad (1)$$

where  $W$  is the length of the sensitive area,  $v$  is the speed of the target vehicle.

In the process of data acquisition, the size of the dynamic sensitive area is judged according to the speed of the target vehicle. The vehicles driving in the dynamic area are assumed to affect the trajectory of the target vehicle.

### 3. MODEL ARCHITECTURE

The model proposed in this paper is shown in Figure 3. It has two embedding layers to map vehicle trajectories and external parameters. The LSTM cell is used to encode the parameters mapped from embedding layers, a convolution layer is proposed to fuse the hidden states of each vehicle, and a decoding layer is designed to output the prediction result of the target vehicle.

#### 3.1 Input and output

If the number of vehicles in the sensitive area of the target vehicle is  $n$ , there are  $n+1$  sets of datasets as the inputs of our model. The aim of the model is to predict the trajectory of the target vehicle at time  $[T_{obs+1}, T_{pred}]$  with trajectory of these vehicles in  $[T_1, T_{obs}]$ . In this paper, the influence of the target vehicle and its surrounding vehicles are considered for trajectory prediction. At the same time, the outline and types of the vehicle and the lane position of the vehicle have a great impact on the trajectory, so the model must take the parameters of these vehicles into account when predicting the vehicle trajectory. Finally, the inputs of this model are:

$$\begin{cases} X = \{X^1, X^2, \dots, X^t, \dots, X^{T_{obs}}\} \\ X^t = \{(x_j^t, y_j^t, p_j) | t \in [1, T_{obs}], j \in [1, n]\} \end{cases} \quad (2)$$

Among them,  $T_{obs}$  is the end point of historical data input;  $X^t$  is the input parameter at time  $t$ ;  $(x_j^t, y_j^t, p_j)$  includes the position coordinate of car  $j$  at time  $t$  and vehicle external parameters;  $n$  is the total number of surrounding vehicles. We defined  $p_j$  in experiment as follow:

$$p_j = [l_{length}, l_{width}, v_{motor}, v_{car}, v_{truck}, t_{left}, t_{right}] \quad (3)$$

where  $l_{length}$  and  $l_{width}$  are the normalised width and length of the vehicles;  $v_{motor}$ ,  $v_{car}$  and  $v_{truck}$  are the expressions of vehicle types which are

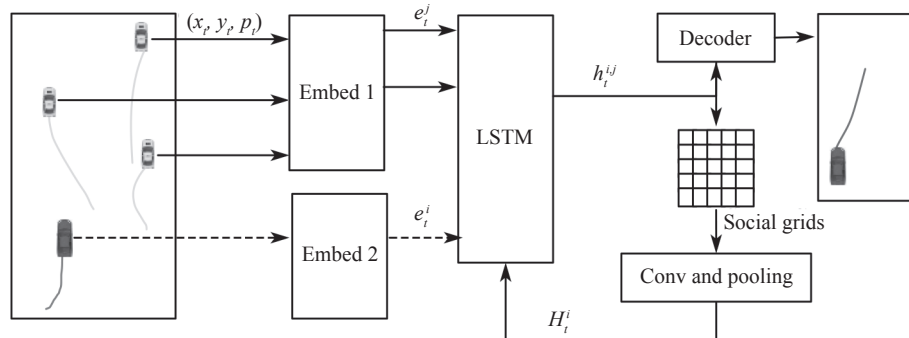


Figure 3 – Model structure

executed by one-hot method;  $t_{left}$  and  $t_{right}$  are the labels for the information of the lane position of the vehicles.

NGSIM provides abundant vehicle data such as type of vehicle, length, width, and inherent parameters of the target vehicle including the lane information and road information.

The output of the model is the trajectory of the target vehicle in the next few seconds:

$$Y = (x_t^i, y_t^i) | t \in [T_{obs} + 1, T_{pred}] \quad (4)$$

According to the experiment [1], the accuracy of predicted result will drop rapidly while the length of sequence exceeds 5 seconds. In order to have the same evaluation system as other models, the longest length of predicted sequence is defined as 5 seconds.

### 3.2 Embedding layer

In order to input the trajectory of surrounding vehicles into a shared LSTM, the data needs to be mapped to the explicit state  $e_t^j$  of common length  $D$ . Another embedding layer is needed for mapping the target vehicle's historical trajectory. Besides, considering the position of target vehicles, the lane and road information are also embedded in  $e_t^i$ :

$$e_t = \Phi(x_t, y_t, p_t, w_e) \quad (5)$$

where  $\Phi$  is a full connect layer,  $w_e$  is the parameter of  $\Phi$ , the shape of  $w_e$  is determined by the length of  $p_t$  and  $D$ . Two different embedding layers are designed for surrounding vehicles and target vehicle separately, because their explicit states  $e_t$  have different information. As is shown in Figure 3,  $e_{t-1}^i$  is generated for target vehicle  $i$  and  $e_{t-1}^j$  is for surrounding vehicle  $j$ .

Our model considers the external parameters of vehicle by adding an additional supplementary parameter  $p_p$ , which consists of types, length width of the vehicles, and the lane information. The specific processing mode is showed in experiment.

### 3.3 LSTM encoder

In the normal driving environment, vehicles change lanes and adjust their speed according to their own conditions including the surrounding vehicle's information, which affects the actions of other vehicles in turn. In this paper, the LSTM cell was used to capture the historical trajectory information of all relevant vehicles. Mutual interference

between the vehicles is also taken into account. Using Social-LSTM as a reference, in this paper we set up one LSTM cell for each vehicle in a scene.

The recessive hidden state  $h_{t-1}^j$  is a vector of length  $D$  obtained by the LSTM after the last cycle. The internal forget gate of the LSTM model is used to control the degree of influence of the recessive features on the trajectory prediction. Explicit state  $e_t^j$  and hidden state  $h_{t-1}^j$  are inputted into the LSTM model after concatenation, and get the hidden state  $h_t^j$  of the car  $j$  at time  $t$ :

$$h_t^j = LSTM(e_t, H_{t-1}^j, h_{t-1}^j) \quad (6)$$

These LSTM cells need to share their hidden state with each other to realise the interaction. Therefore, a shared convolution layer is designed for the model to fuse all hidden states from LSTM cells.

After obtaining the hidden states of all surrounding vehicles, a function named  $l_{mn}$  is proposed to fuse hidden states in order to obtain environment characteristics of the target vehicle  $H_t^i(m, n)$ . For the next step, a convolution layer and a pooling layer are used to shrink the data scale to  $D$ , which preserves position relationship of vehicles according to the used region proposal network (RPN) proposed by He [23].

$$H_t^i(m, n) = l_{mn} \cdot \sum_{j \in N_i} (x_t^j - x_t^i, y_t^j - y_t^i) \cdot h_t^j \quad (7)$$

where  $N_i$  is the set of surrounding vehicles; the function  $l_{mn}$  judges whether the car  $j$  is in the  $(m, n)$  grid. The obtained  $H_t^i$  is the surrounding feature map, which contains all the historical trajectory features of surrounding vehicles.  $H_t^i$  is used to input LSTM for the next round of trajectory prediction and judgment.

### 3.4 Prediction

In order to obtain the points of prediction trajectory, the hidden states  $h_t^j$  generated by the LSTM need to be mapped by a decoding layer showed in Figure 3. Since the reliability of the prediction results is still poor, considering Alahi's advice [17], in this paper our model outputs a bivariate normal distribution to obtain the location area of target car at the next time point, which includes the relevant parameters of  $N(\mu_{t+1}^i, \sigma_{t+1}^i, \rho_{t+1}^i)$ :

$$\mu_{t+1}^i = (\mu_x, \mu_y)_{t+1}^i, \sigma_{t+1}^i = (\sigma_x, \sigma_y)_{t+1}^i \quad (8)$$

where  $\mu_x$  and  $\mu_y$  are the means of bivariate normal distribution, the  $\sigma_x$  and  $\sigma_y$  are the standard deviations of the same Gaussian distribution. The  $\rho$  is the correlation parameter of two dimensions.

The specific mapping relationship is:

$$N_{para} = \psi(h_t^i; w_N) \tag{9}$$

where  $N_{para} = [\mu_{t+1}^i, \sigma_{t+1}^i, \rho_{t+1}^i]$ ,  $\psi$  is a full connect layer to embed hidden states to  $N_{para}$ ,  $w_N$  represents the parameters of this layer.

Owing to the accuracy of the model and vehicles tracks generally appearing on the road, a constraint condition is added in the output layer of the model: the last layer of the mapping uses the sigmoid activation function to constrain the output of the vehicles' horizontal coordinates to (0,1) in order to avoid the output exceeding the roadside range.

### 4. EXPERIMENT

In this section, our model is trained and evaluated in the NGSIM dataset. Meanwhile, the accuracy and efficiency of our model are compared with other models.

#### 4.1 Data pre-processing

We selected Lankershim Boulevard and US-101 road data in the NGSIM dataset as experimental data. As shown in Figure 4, the sections from Lankershim Boulevard without any bends and traffic lights were selected for the experiment. For the US-101 road, we also took the straight section for the research.

During data filtering, 50,000 sequences of US-101 datasets and 50,000 sequences of Lankershim Boulevard datasets were obtained by filter. Each sequence includes the target vehicle and surrounding vehicles which are in a rectangle area. A sequence of selected sample is shown in Figure 5:

Each training and testing sequence is processed in the following steps:

- 1) Downsampling  
The original data are collected at 10 Hz, which cause too much load for our model. After multiple experiments, the sampling frequency is set to 2 Hz, which is accurate enough to express a trajectory of vehicle.
- 2) Filtering useless data  
NGSIM dataset contains more than 25 columns including Vehicle ID, Frame ID, Global Time. We filter some columns which are useless for the trajectory prediction out. Meanwhile, the following processes could run faster.
- 3) Setting an area to get sequence  
After setting a random starting point and starting time in global coordinates, the information of vehicles driving in this area at this moment is recorded in the next 10 seconds. The sequences with data missing will be abandoned.
- 4) Normalisation  
The biggest problem is the universality of the model. However, the differences in coordinates of different sequences are tremendous. Therefore, a normalisation method is designed for all of the coordinates in this paper as follows:

$$\begin{cases} \tilde{x}_k = \frac{x_k}{w_{road}} \\ \tilde{y}_k = \frac{y_t - y_{min}}{y_{max} - y_{min}} \end{cases} k \in [0, 19] \tag{10}$$

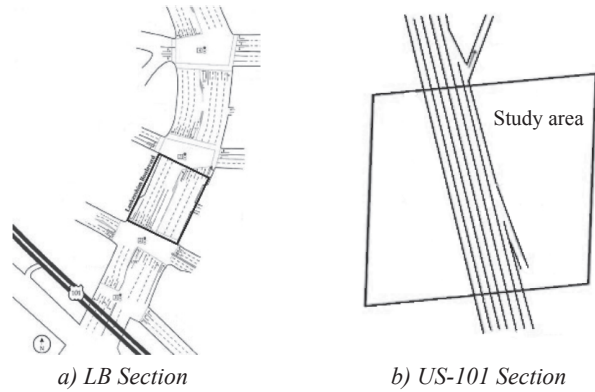


Figure 4 – Research section

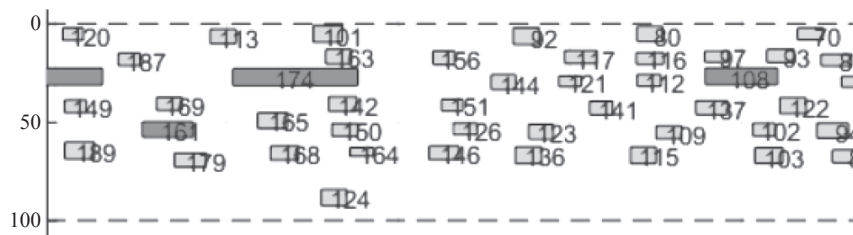


Figure 5 – US-101 data sample

where  $(x_k, y_k)$  is the global coordinate of the vehicles.  $(\tilde{x}_k, \tilde{y}_k)$  is the normalised coordinate.  $y_{max}$  and  $y_{min}$  are the maximum and minimum ordinates in current sequence.

The vehicle types are identified by number 1~3 in NGSIM, our model encoded it through one-hot (Table 1).

Table 1 – Coding methods of vehicle types

Vehicle types	NGSIM	One Hot
Motor	1	001
Car	2	010
Truck	3	100

Besides, the length and the width of vehicles are also encoded by min-max normalisation. One-hot method is also applied in lane information execution (Figure 6).

Finally, the sliding window method is used to gather the short sequence of length  $n$  from the trajectory sequence into LSTM. The first 5 seconds of the sequence is used as the historical trajectory and the last 5 seconds of data is used as the future trajectory. The sample is shuffled and divided into 8:2 for training and testing. In addition, a dataset including trucks is set to test the stability of the model.

## 4.2 Parameter adjustment and evaluation

For the model's parameters, we set the input vector length  $D$  as 32 and Adam is used as optimiser with the batch size of 128. Learning rate decay is used to accelerate the decline of loss in the early stage and make loss smaller latterly. Therefore, learning rate reduces 0.05 per 5-epochs during training and the initial value was 0.5. Besides, the training param-

eters are initialised with normal random numbers. For the dataset, the trajectory data is normalised by min-max. We selected the external parameter as vehicle types (motorcycles, cars, trucks), vehicle length, and vehicle width, combined with the vehicle lane and road information. One-hot method is applied in the execution of the vehicle types, which eventually export encoded characters.

The loss function of this model includes three parts. The first part is the target vehicle position prediction deviation obtained by negative log likelihood (NLL) loss:

$$L_1^i = -\frac{1}{T_{pred} - T_{obs}} \cdot \sum_{t=T_{obs}+1}^{T_{pred}} \log(P(x_t^i, y_t^i | N_{para})) \quad (11)$$

where  $P(\cdot)$  is the likelihood function.  $T_{pred}$  is the last point of prediction sequence.  $T_{obs}$  is the current time.  $N_{para}$  is the parameter of bivariate normal distribution obtained by the model. Equation 7 explains the specific parameters of  $N_{para}$ . The purpose of  $L_1^i$  is making the bivariate normal distribution as close to the real value as possible.

The second part is the prediction deviation of the hidden parameters of surrounding vehicles. Different from other models, the target vehicle and surrounding vehicles share the same LSTM cell, so the hidden parameters of all the vehicles can be used as the loss value to train the LSTM. Setting this deviation can make full use of the dataset and train the LSTM parameters better:

$$L_2^i = \frac{1}{T_{pred} - T_{obs}} \cdot \sum_{j \in N_j} \frac{1}{n} \sum_{t=T_{obs}+1}^{T_{pred}} \log(P(x_t^j, y_t^j | N_{para}^j)) \quad (12)$$

where  $N_{para}^j = \psi(out_t^j; w_N)$  is the normal parameter of trajectory prediction of surrounding vehicle  $j$ .  $n$  is the total number of the surrounding vehicles.

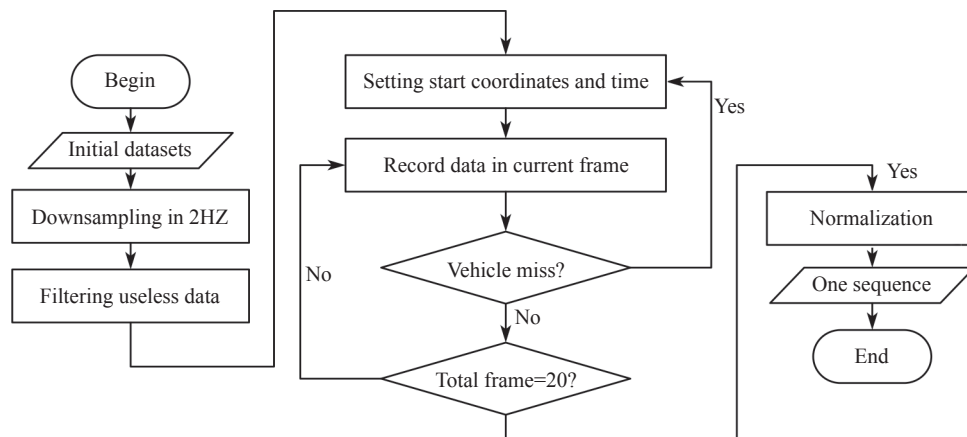


Figure 6 – Dataset execution

This paper also quotes the mean square error (MSE) loss function used in models such as Convolution Social LSTM. Since the outputs of the model are the parameters of a bivariate normal distribution, the parameters of bivariate normal distribution  $(\mu_x, \mu_y)$  are selected as the prediction point. The final RMSE loss is as follows:

$$L_{RMSE} = \frac{1}{T_{pred} - T_{obs}} \sum_{t=T_{obs}+1}^{T_{obs}+T_{pred}} [(x_t - \mu_t)^2 + (y_t - \mu_t)^2] \quad (13)$$

During training, the loss function taken in this paper is  $L=L_1^i+L_2^i+L_{RMSE}$ . Meanwhile, the same evaluation system is selected, namely NLL loss  $L_1^i$  and RMSE loss  $L_{RMSE}$ .

### 4.3 Training and fine-tuning

Training mode is firstly applied for pre-training. Sequences are cut into two sub sequences for training, and the length of each is 19 (Figure 7).

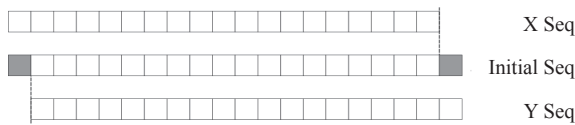


Figure 7 – Segmentation training method

In order to improve long-term prediction accuracy, our model is fine-tuned in the same sequences with different segmentation (Figure 8).

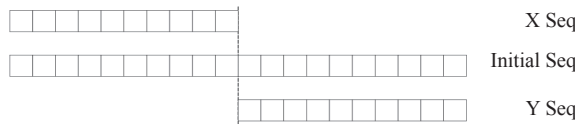


Figure 8 – Segmentation fine-tuning method

A sequence contains 20 points. In training mode, we use  $t$  point and history information to predict  $t+1$  point. In fine-tuning mode, the 10 points ahead are used to predict the last 10 points.

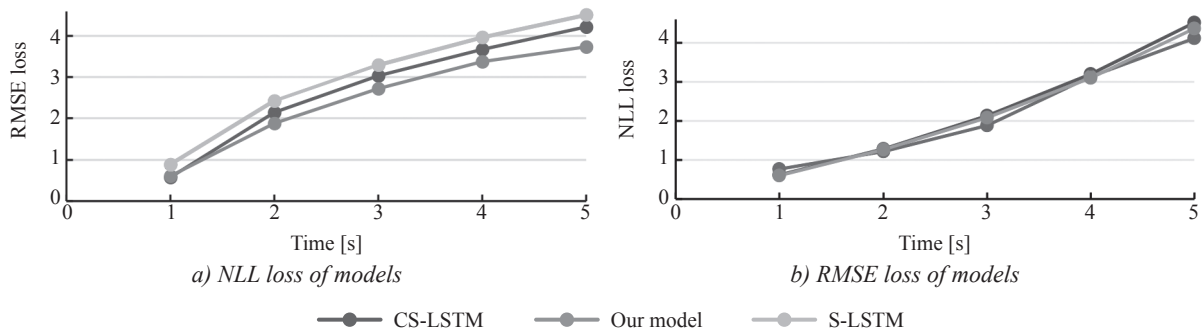


Figure 10 – Comparison results of different models

### 4.4 Performance analysis of dynamic sensitive area

In order to verify the effect of dynamic sensitive areas, we set up static sensitive areas in the experiment, which does not filter the sequence and takes all vehicles within 150 m around the target vehicles as input. Comparing with the models using dynamic sensitive area, we obtained the loss function after smoothing (Figure 9).

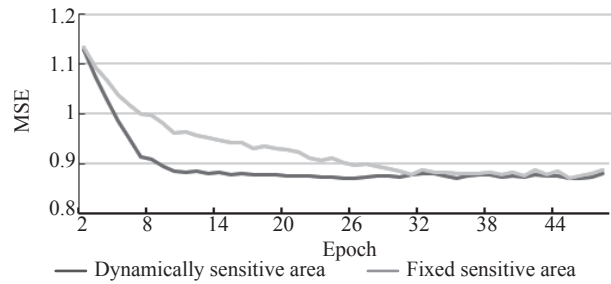


Figure 9 – MSE of model with different area

Since the data can be reasonably filtered by the dynamic sensitive area, the decline speed of the training process has been greatly improved. Without the dynamic sensitive area, the time required for each batch is 0.633~1.102 seconds, which drops to 0.352~0.836 seconds after using dynamic regions. At the same time, the shortened input sequence reduces the iteration speed exponentially. Therefore, the dynamic sensitive area proposed in this paper has great improvement of training speed and real-time performance.

### 4.5 Predictive accuracy analysis

We select 80,000 sets of driving data to train and use 20,000 sets to evaluate the accuracy of the model in the experiment. The final results of comparison with CS-LSTM and Social LSTM (S-LSTM) are shown in the Table 2. The comparison standard is root mean square error (RMSE) [14] and log likelihood loss (NLL) (Figure 10).

Table 2 – RMSE and NLL of models

Assessment method	Time [s]	S-LSTM	CS-LSTM	Our model	Truck as object
RMSE	1	0.61	0.62	0.61	0.58
	2	1.27	1.29	1.27	1.13
	3	2.09	2.13	2.03	2.10
	4	3.10	3.20	2.82	2.33
	5	4.37	4.52	3.37	3.17
NLL	1	0.89	0.58	0.77	0.70
	2	2.43	2.14	2.13	2.10
	3	3.30	3.03	2.89	3.13
	4	3.97	3.68	3.13	3.01
	5	4.51	4.22	4.11	3.98

The RMSE loss of our model gradually exceeds other models with time. This is because the input sequence contains more important fixed information. During long-term prediction, these parameters have a greater impact on the vehicle trajectory. The accuracy of the model at 5 seconds is 23.7% higher than in other models. The last column in Table 2 is the experimental result, where the truck is used as the target vehicle. Meanwhile, owing to considering the possible impact of vehicle external parameters in the input sequence, the loss value of our model has a lower decline compared to other models.

In order to verify the stability of our model with the interference of external parameters, this paper selects four sets of data which have different conditions including vehicle types and lane information. CS-LSTM is a state-of-the-art model for predicting trajectory which ignores the above parameters. The parameters of bivariate normal distribution

$(\mu_x, \mu_y)$  are selected as the predictive point. The predictive trajectory is shown in Figure 11, where circle points represent the true trajectory, triangle points are the predictive trajectory of our model, and square points are the CS-LSTM result.

Figure 11a is the normal condition in which the prediction result is not affected by the vehicle types and lane condition. It can be seen that the results of the two models are close to the true trajectory.

Figure 11b is the condition that the target vehicle hugged the side of the road. With ignoring lane information, the prediction result of the driving intention is wrong for CS-LSTM. In contrast, considering the road information, our model provides a relatively accurate result.

Figure 11c is the condition that a truck is one of surrounding vehicles. The length and width of the truck are commonly bigger than those of other types of vehicle. Therefore, there must be more distance when other cars overtake a truck. The prediction

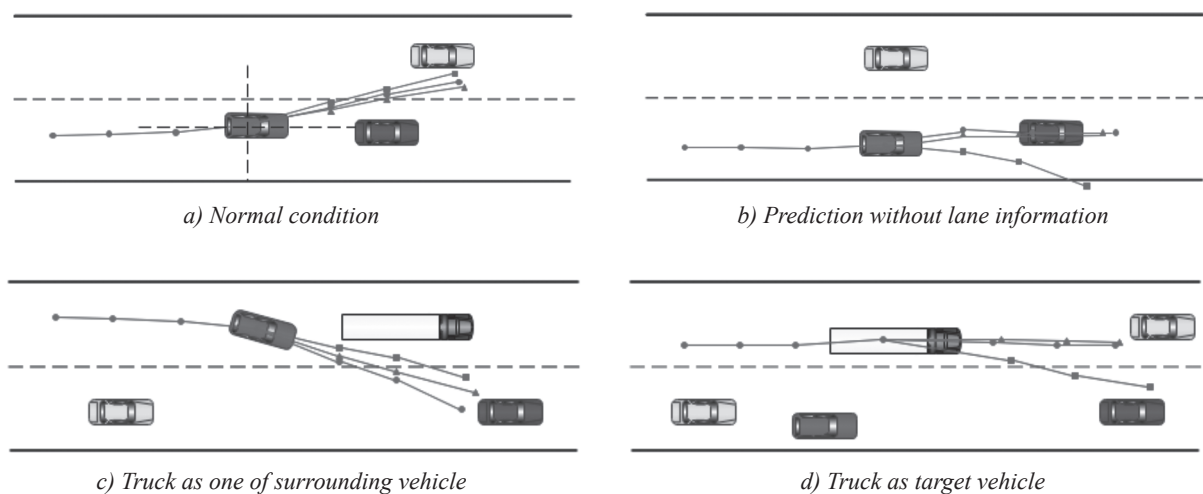


Figure 11 – Vehicle trajectory prediction under the different conditions



Table 3 – Target vehicle trajectory prediction with truck participation

	True coordinates [m]	S-LSTM	Our model
P1	(9.8, 2.1)	(9.7, 1.9)	(9.7, 2.2)
P2	(11.2, 4.7)	(11.2, 3.7)	(10.8, 4.7)
P3	(13.8, 9.3)	(11.7, 8.2)	(12.8, 10.0)

result of our model is obviously closer to the truth and safer than the CS-LSTM result. Specific data including true value, prediction results of the S-LSTM, and our model are given in Table 3. The maximum accuracy gap between the two models exceeds 25.6%.

Figure 11d shows the result with the truck being the target vehicle. Trucks usually have lower maximum acceleration than cars, so the necessary conditions of overtaking are more stringent. Without considering the types of vehicle, the CS-LSTM finally has a negative prediction result of driving intention. Meanwhile, the predictive line of our model can track the true trajectory accurately.

In conclusion, the model proposed in this paper comprehensively considers the impact of vehicle external parameters on the trajectory. The results show that our model can avoid some prediction errors of driving intention and have better accuracy of prediction result.

## 5. CONCLUSION

Aiming at the fact that the existing data-driven vehicle trajectory prediction algorithm cannot consider the impact of vehicle external parameters, this paper proposes a vehicle trajectory prediction model that can handle the vehicle external parameters. The historical trajectory and the external parameters of the vehicles are used as inputs, and the bivariate normal distribution parameters of the vehicles prediction are eventually obtained. In our model, a LSTM cell is designed to encode the historical trajectory and a convolution layer is used to fuse hidden state of all vehicles so that our model can predict trajectory of the target vehicle considering the influence of the surrounding vehicles.

Meanwhile, the existing algorithms ignore the difference between the longitudinal and lateral safe following distance when the vehicles are driving normally. Therefore, the dynamic sensitive area

based on vehicle speed is proposed in this paper, which is used to filter other vehicles around the target vehicle.

Finally, the model is trained and evaluated in the NGSIM data. The results show that the predictive accuracy of our model is greatly improved by 23.7% when compared to other models in long-term prediction. The model's ability of anti-interference with the external parameters such as vehicle types is also improved. The results show that our model can avoid some errors of predicting driving intention and have better accuracy of prediction result. Furthermore, the dynamic sensitive area proposed in this paper is significantly optimised to speed up the training process. The time required for each batch with a normal sensitive is 0.633~1.102 seconds, which drops to 0.352~0.836 seconds after using dynamic regions.

## ACKNOWLEDGEMENT

The work presented in this paper was supported by the National Key R&D Program of China: “Key Technologies and Equipment for Intelligent Monitoring and Early Warning of Social Security Events” (2018YFC0807500).

李旭川, 硕士<sup>1,2</sup>

电子邮件: Rayne\_Sun@outlook.com

范林坤, 硕士<sup>1,2</sup>

电子邮件: shenywl.think@outlook.com

陈涛, 博士<sup>1,2</sup>

(通讯作者)

电子邮件: Chentao@chd.edu.cn

郭丛帅, 硕士<sup>1,2</sup>

电子邮件: 2578729340@qq.com

<sup>1</sup> 交通运输部汽车运输安全技术重点实验室

长安大学, No. 126, 南二环中段, 西安, 中国

<sup>2</sup> 汽车学院, 长安大学, No. 126, 南二环中段, 西安, 中国

## 考虑车辆固有参数的轨迹预测模型

对自动驾驶汽车来说, 预测周围车辆轨迹是不可少的。针对现有模型不能充分利用车体轮廓、车道信息等外部参数的问题, 我们提出了一种能够充分利用这些参数预测直道非自由流状态下的周车轨迹预测模型。此外, 我们提出了动态敏感区域, 用以过滤不重要的周车。模型以车辆的历史轨迹和固有参数作为模型输入, 还提出了一个共享的长短期记忆胞元来映射历史轨迹和外部参数, 最后得到隐藏状态。接着, 车辆隐藏状态用于提取潜在驾驶意图。最后, 我们设计了一个卷积模块来融合隐藏状态, 以提供给下一个预测循环。本文设计了解码器, 用以对目标车辆的隐藏状态进行解码, 得到预测结果。实验结果表明, 本文提出的动态敏感区训

练算法可以将训练时间缩短到未使用时的的75.86%。与其他较为优秀的模型相比，该模型的精度提高了23.7%。同时，模型相对于外界参数的抗干扰能力有很大的提高。

#### 关键词

轨迹预测；车辆外部参数；动态敏感区域；长短期记忆网络。

#### REFERENCES

- [1] Deo N, Trivedi MM. Convolutional social pooling for vehicle trajectory prediction. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2018;1805: 1468-76. DOI: 10.1109/CVPRW.2018.00196
- [2] Ji X, et al. Driving intention recognition and vehicle trajectory prediction based on LSTM network. *Journal of China Highway*. 2019;32(6): 34-42.
- [3] Barth A, Franke U. Where will the oncoming vehicle be the next second? *2008 IEEE Intelligent Vehicles Symposium, 4-6 June 2008, Eindhoven, The Netherlands*; 2018. p. 1068-1073. DOI: 10.1109/IVS.2008.4621210
- [4] Toledo-Moreo R, Zamora-Izquierdo MA. IMM-based lane-change prediction in highways with low-cost GPS/INS. *IEEE Transactions on Intelligent Transportation Systems*. 2009;10(1): 180-5. DOI: 10.1109/TITS.2008.2011691
- [5] Schubert R, et al. Empirical evaluation of vehicular models for ego motion estimation. *2011 IEEE Intelligent Vehicles Symposium (IV), 5-9 June 2011, Baden-Baden, Germany*; 2011. p. 534-539. DOI: 10.1109/IVS.2011.5940526
- [6] Houenou A, Bonnifait P, Cherfaoui V, Yao W. Vehicle trajectory prediction based on motion model and maneuver recognition. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 3-7 Nov. 2013, Tokyo, Japan*; 2013. p. 4363-4369. DOI: 10.1109/IROS.2013.6696982
- [7] Singh K, Li B. Estimation of traffic densities for multilane roadways using a Markov model approach. *IEEE Transactions on Industrial Electronics*. 2012;59(11): 4369-76. DOI: 10.1109/TIE.2011.2180271
- [8] Berndt H, Emmert J, Dietmayer K. Continuous driver intention recognition with hidden Markov models. *2008 11<sup>th</sup> International Conference on Intelligent Transportation Systems, 12-15 Oct. 2008, Beijing, China*; 2008. p. 1189-1194. DOI: 10.1109/ITSC.2008.4732630
- [9] Huang L, Guo H, Zhang R, Wu J. Lane-changing behavior model of unmanned vehicles based on LSTM in human-machine hybrid driving environment. *Journal of China Highway*. 2020;33(07): 156-66.
- [10] Tang J, et al. Lane-changes prediction based on adaptive fuzzy neural network. *Expert Systems with Applications*. 2018;91: 452-63. DOI: 10.1016/j.eswa.2017.09.025
- [11] Wissing C, Nattermann T, Glander K, Bertram T. Probabilistic time-to-lane-change prediction on highways. *2017 IEEE Intelligent Vehicles Symposium (IV), 11-14 June 2017, Los Angeles, CA, USA*; 2017. p. 1452-1457. DOI: 10.1109/IVS.2017.7995914
- [12] Zyner A, Worrall S, Ward J, Nebot E. Long short term memory for driver intent prediction. *2017 IEEE Intelligent Vehicles Symposium (IV), 11-14 June 2017, Los Angeles, CA, USA*; 2017. p. 1484-1489. DOI: 10.1109/IVS.2017.7995919
- [13] Phillips DJ, Wheeler TA, Kochenderfer MJ. Generalizable intention prediction of human drivers at intersections. *2017 IEEE Intelligent Vehicles Symposium (IV), 11-14 June 2017, Los Angeles, CA, USA*; 2017. p. 1665-1670. DOI: 10.1109/IVS.2017.7995948
- [14] Kuefler A, Morton J, Wheeler TA, Kochenderfer MJ. Imitating driver behavior with generative adversarial networks. *2017 IEEE Intelligent Vehicles Symposium (IV), 11-14 June 2017, Los Angeles, CA, USA*; 2017. p. 204-211. DOI: 10.1109/IVS.2017.7995721
- [15] Liu Z, Wu X, Ni J, Zhang T. Driving intention recognition based on the cascade algorithm of HMM and SVM. *Automotive Engineering*. 2018;40(07): 858-64.
- [16] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*. 1997;9(8): 1735-80. DOI: 10.1162/neco.1997.9.8.1735
- [17] Alahi A, et al. Social LSTM: Human trajectory prediction in crowded spaces. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 27-30 June 2016, Las Vegas, NV, USA*; 2016. p. 961-971. DOI: 10.1109/CVPR.2016.110
- [18] Liu C, Liang J. Vehicle trajectory prediction based on attention mechanism. *Journal of Zhejiang University (Engineering Science Edition)*. 2019;14: 1-8.
- [19] Wang Y, Yang S, Pan B. Research on vehicle trajectory of highway straight section. *Highway Traffic Technology*. 2016;33(02): 111-9.
- [20] Spacek P. Track behavior in curve areas: Attempt at typology. *Journal of Transportation Engineering*. 2005;131: 669-76. DOI: 10.1061/(ASCE)0733-947X(2005)131:9(669)
- [21] Institution CS. GB/T 51328-2018. *Stand for urban comprehensive transport system planning*. Ministry of Housing and Urban-Rural Development, PRC; 2003. p. 33-41.
- [22] NGSIM: *Next generation simulation*. In: DataHub I; 2016.
- [23] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Neural Information Processing Systems*. 2015: 91-9.