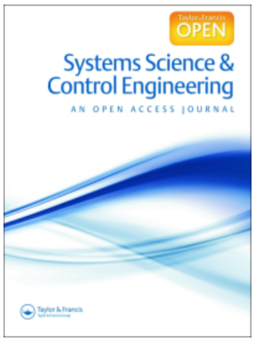


Systems Science & Control Engineering

An Open Access Journal

ISSN: 2164-2583 (Online) Journal homepage: www.tandfonline.com/journals/tssc20

Vehicle trajectory prediction based on attention optimized with real-scene sampling

Zhiyu Yang, Yunlong Wan, Li Du, Wei Zhang, Xue Yang & Yunwu Han

To cite this article: Zhiyu Yang, Yunlong Wan, Li Du, Wei Zhang, Xue Yang & Yunwu Han (2024) Vehicle trajectory prediction based on attention optimized with real-scene sampling, Systems Science & Control Engineering, 12:1, 2347889, DOI: 10.1080/21642583.2024.2347889

To link to this article: https://doi.org/10.1080/21642583.2024.2347889

Published online: 15 May 2024.

SYSTEMS SCIENCE & CONTROL ENGINEERING: AN OPEN ACCESS JOURNAL
2024, VOL. 12, NO. 1, 2347889
https://doi.org/10.1080/21642583.2024.2347889

Taylor & Francis
Taylor & Francis Group

Vehicle trajectory prediction based on attention optimized with real-scene sampling

Zhiyu Yang[a], Yunlong Wan[a], Li Du[a], Wei Zhang[a], Xue Yang[a] and Yunwu Han[b]

[a]School of Mechanical Engineering, Chongqing Technology and Business University, Chongqing, People's Republic of China; [b]School of Intelligent Transportation, Jiangsu Vocational College of Electronics and Information, Huaian, People's Republic of China

ABSTRACT

Advancements in autonomous vehicles and deep learning have notably improved vehicle trajectory prediction accuracy. However, extracting interaction features in complex driving scenarios, such as vehicle-to-vehicle interactions and lane constraints, presents challenges. Deep learning-based methods struggle to achieve optimal predictive performance under limited computational resources. This study introduces a global attention mechanism to enhance feature extraction from driving scene encodings, focusing the decoder on interactive behaviours and boosting long-term prediction performance. An adaptive scheduled sampling model is employed, using actual driving scenarios probabilistically for training, addressing slow learning of actual driving behaviours and lack of initial feature correction. This method increases attention to actual interactions, reducing reliance on natural scenes and improving model generalizability. On the NGSIM dataset, sampling attention encoder-decoder (SAED) achieves a 1–5 s average displacement error (ADE) of 1.34 m, with 4 s and 5 s final displacement errors (FDEs) of 1.64 and 2.06 m, respectively. Compared to methods based on long short-term memory (LSTM), SAED reduces the model's storage space by 24.68% under the same network layer count. That demonstrates its effectiveness in extracting interactive behaviours in complex scenarios and enhances the accuracy of long-term predictions.

ARTICLE HISTORY
Received 18 January 2024
Accepted 22 April 2024

KEYWORDS
Trajectory prediction; autonomous vehicles; deep learning

1. Introduction

Trajectory prediction, as one of the essential tasks of autonomous driving, requires accurate prediction of the target agent's position in the next few seconds. Due to the complex interaction behaviours between vehicles that change in real-time, road constraints on vehicle trajectories, and the vehicle's kinematic limitations, it is difficult for interaction-based trajectory prediction methods that require artificial rules to have desired trajectories. In order to fully extract the potential interaction information in driving scenarios and achieve high accuracy long-term prediction missions, it is necessary to model the interaction behaviours in complex driving scenarios (Gindele et al., 2010; Lefèvre et al., 2013).

For the complex interaction features of the target vehicle, since it is difficult for dynamics and kinematics methods to represent the interaction behaviour fully, it is necessary to extract the temporal and spatial correlations in the data using deep learning-based methods. For example, Deo and Trivedi (2018) combined a convolutional social pooling network with the LSTM to predict vehicle trajectories on a highway. Altché and de La Fortelle (2017) used the LSTM to predict the longitudinal speed of the vehicle on a highway segment while considering the trajectories of nine vehicles around this vehicle. Kim et al. (2017) proposed an LSTM-based trajectory prediction method using the occupancy grid maps to describe the driving environment. Lee et al. (2017) proposed a recurrent neural network (RNN) encoder-decoder framework for deep stochastic inverse optimal control, which can predict the trajectories of interacting road users in dynamic scenarios. Although many studies have built neural network models individually to extract potential features in interactions, different interaction features affect interactions differently, and these approaches still have limitations.

Attention mechanisms are currently gaining popularity in deep learning, which can compute interaction features to highlight interaction behaviours and provide assistance in improving the long-term prediction of trajectory prediction models compared to commonly used neural network approaches. Using the probabilistic approach, Tang and Salakhutdinov (2019) constructed a multi-trajectory prediction model with an end-to-end

CONTACT Yunwu Han ✉ hanyw08@jsei.edu.cn

structure The model consists of an RNN with a set of parallel shared parameters, and a dynamic encoder based on the attention mechanism is constructed to learn diverse information. Then, the probability of multi-trajectories is decoded and computed to obtain the predicted trajectory. In the work of Mercat et al. (2020), the multi-model trajectory predictions are performed through the multi-head attention layer, which uses LSTM as an encoder-decoder, and two attention layers are equipped in the middle layer. Messaoud, Yahiaoui, et al. (2021) modelled the interactions between traffic flow participants by extracting attention from the LSTM encoder. Based on this work, Messaoud, Deo, et al. (2021) set up multiple attention heads, and each attention head can model the possible interaction behaviours between the target and the contextual features.

Additionally, while long short-term memory (LSTM) networks have been extensively applied in trajectory prediction, their complex structure and computational cost present limitations, especially in autonomous driving systems where rapid response is crucial. In other domains, gated recurrent units (GRUs) have been demonstrated to outperform LSTMs in both performance and computational efficiency (Yamak et al., 2020; Yang et al., 2020; Zhang et al., 2021).

Consequently, this study introduces a novel deep learning approach named sampled attention encoder-decoder (SAED), based on an encoder-decoder architecture, aimed at addressing the challenge of balancing predictive performance and computational demands in long-term forecasting inherent in traditional deep learning methods.

Our contributions are highlighted as follows:

(1) To tackle the issue of decreased long-term prediction accuracy due to the difficulty in extracting complex interactive behaviours in driving scenarios, this paper establishes a global attention model within the encoder-decoder framework. This model aids in processing both vehicle-to-vehicle interactions and the influence of lane constraints within driving scenes. The model's long-term predictive performance is enhanced by enabling the decoder to focus on primary interactions while disregarding secondary ones.
(2) In trajectory prediction, models often require multiple iterations of training to learn realistic driving behaviours, thereby improving the accuracy of predicted trajectories. This research introduces an adaptive scheduled sampling method that probabilistically corrects the historical trajectory encodings obtained by the encoder with actual trajectory data. This accelerates the model's learning of authentic driving behaviours, enhancing long-term prediction accuracy and reducing the number of iterations needed to achieve comparable predictive performance.

2. Problem statement

In this section, we analyze the formulation of trajectory prediction problems in interactive scenarios and describe the dataset used and its preprocessing.

2.1. Problem formulation

In this study, the trajectory prediction of vehicles is considered as a sequence generation problem. The future driving trajectories of the target vehicle are predicted based on the past global driving features of the target vehicle observed in the divided time T. The global driving features in the driving scene observed by the target vehicle during the divided time are used to predict the potential trajectories of the target vehicle in the future. The driving features of the target vehicle during $[0, T]$ in the dataset and the spatiotemporal features based on the interaction behaviours in the driving scenario are used as the input data, and the total input is $\tilde{X} = \{S, v, \mathrm{a}, \Delta S, \Delta v, \Delta \mathrm{a}\}$. The driving features of the vehicle include the 2D position $S = [(x^0, y^0), (x^1, y^1), \ldots, (x^t, y^t)]$, transverse and vertical velocities $v = [(v_x^0, v_y^0), (v_x^1, v_y^1), \ldots, (v_x^t, v_y^t)]$, and transverse and longitudinal accelerations $a = [(a_x^0, a_y^0), (a_x^1, a_y^1), \ldots, (a_x^t, a_y^t)]$. The spatiotemporal features based on the interaction behaviour include the 2D position $\Delta S = [(\Delta x_{\mathrm{ei}}^0, \Delta y_{ei}^0), (\Delta x_{\mathrm{ei}}^1, \Delta y_{ei}^1), \ldots, (\Delta x_{\mathrm{ei}}^t, \Delta y_{ei}^t)]$ of the surrounding vehicle (i) relative to the target vehicle (e), the absolute transverse and longitudinal velocities $v^i = [(v_x^{i,0}, v_y^{i,0}), (v_x^{i,1}, v_y^{i,1}), \ldots, (v_x^{i,t}, v_y^{i,t})]$ of the surrounding vehicle (i), the absolute transverse and longitudinal accelerations $a^i = [(a_x^{i,0}, a_y^{i,0}), (a_x^{i,1}, a_y^{i,1}), \ldots, (a_x^{i,t}, a_y^{i,t})]$ of the surrounding vehicle (i), and the road scene identifiers R_l, R_r (which represent whether the lane where the target vehicle contains a left or right lane; R_l, R_r is 1 if there is a left (right) lane, and 0 if there is not).

The vehicles with relative longitudinal displacement to the target vehicle (e) within [−100 m, 100 m] are chosen as the surrounding vehicles (i), including the six nearest traffic participants in the lane where the target vehicle is as well as its left lane and right lane. In this study, the 44-dimensional global driving features are used as the input data, including the driving features S, v, a of the target vehicle in the global traffic scene and the spatiotemporal features ΔS, v^i, a^i, R_l and R_r based on interactions in the global driving scene. The detailed description of the 44-dimensional features is shown in Table 1.

Table 1. Detailed description of the 44-dimensional features.

Driving feature	Feature description	Dimension
S	Lateral and longitudinal coordinates of the target vehicle (x, y)	6
v	Lateral and longitudinal velocities of the target vehicle (v_x, v_y)	
a	Lateral and longitudinal accelerations of the target vehicle(a_x, a_y)	
ΔS	Relative lateral and longitudinal coordinates of the target vehicle(e) with respect to surrounding vehicle(i)$(\Delta x_{ei}, \Delta y_{ei})$	36
v^i	Relative lateral and longitudinal velocities of the target vehicle with respect to surrounding vehicle(i)(v_x^i, v_y^i)	
a^i	Relative lateral and longitudinal accelerations of the target vehicle with respect to surrounding vehicle(i)(a_x^i, a_y^i)	
R_l, R_r	Lane marker flags for the target vehicle's left and right lanes: 1 if they exist, 0 otherwise	2

Where, S, v and a are divided into horizontal and vertical features, totalling 6 dimensions; similarly, ΔS, v^i and a^i include the interaction features of the target vehicle with six surrounding vehicles, totalling 36 dimensions; R_l and R_r are two dimensions. The global features over the entire time are input using a sliding window, and the past trajectories are used to generate the vehicle trajectories $\tilde{Y} = [(x^{t+1}, y^{t+1}), (x^{t+2}, y^{t+2}), \ldots, (x^{t+T}, y^{t+T})]$ in the future.

2.2. Dataset description and preprocessing

To investigate trajectory prediction tasks in interactive scenarios, it is crucial to select datasets rich in vehicle-to-vehicle interactions and lane constraints for experimentation. The Next Generation Simulation (NGSIM) dataset, developed by the United States Federal Highway Administration, is employed for research in autonomous driving (U.S. Department of Transportation - FHWA, 2022). This dataset encompasses vehicle trajectory data from the US-101 highway and the I-80 interstate, with a schematic of the roads presented in Figure 1. The illustration indicates that the dataset includes an extensive array of lane constraints, meeting the criteria of this study.

This paper aims to investigate the driving behaviour of the target vehicle within 8 s. The sampling frequency of the dataset is 10 Hz, each path is divided into multiple sequences of length 80 frames, and the sliding window has a step size of 1 frame. The raw data of the dataset were collected by cameras located at different sections of the road and were processed into the dataset used in this study after extraction. Therefore, there are certain errors in the dataset, especially the errors in lateral velocity are more significant. This study employs the wavelet decomposition method to filter the dataset. The dataset includes the following maneuvre manners: lane keeping (straight ahead), change to the right lane, and change to the left lane. The driving behaviour is labelled according to the lane change: change to left lane denoted as 0, lane keeping denoted as 1, and change to right lane denoted as 2. According to the study of Y. Zhang et al. (2022), the changes in heading angles are small in the NGSIM dataset due to the high speed. Therefore, a fixed steering angle θ_{Yaw} is set, and the intersection point between the vehicle's trajectory and the lane line is P. When the target vehicle changes lanes, the steering angle of the time step approaching point P is larger. Therefore, preprocessing of the dataset is required to ensure that the selected trajectory sequence contains the complete lane changing behaviour. Intuitively, during an ideal lane change, the vehicle's heading angle will first increase, and then, as the vehicle approaches the lane line, the heading angle begins to decrease. Therefore, to ensure the acquisition of trajectory sequence data that includes the complete lane-changing behaviour, we assess the vehicle's heading angle and consider the part of the trajectory before and after crossing the lane line as part of the lane-changing action. At three moments forward or backward from point P, if the heading angle θ satisfies $|\theta| < \theta_{\text{Yaw}}$, the third moment forward is characterized as the moment T_{Start} to start changing lanes, and the third moment backward is characterized as the moment T_{End} when the lane change ends. The trajectory of $[T_{Start}, T_{End}]$ is denoted as the lane change process.

Table 2. Sample numbers of different maneuvres in the data set.

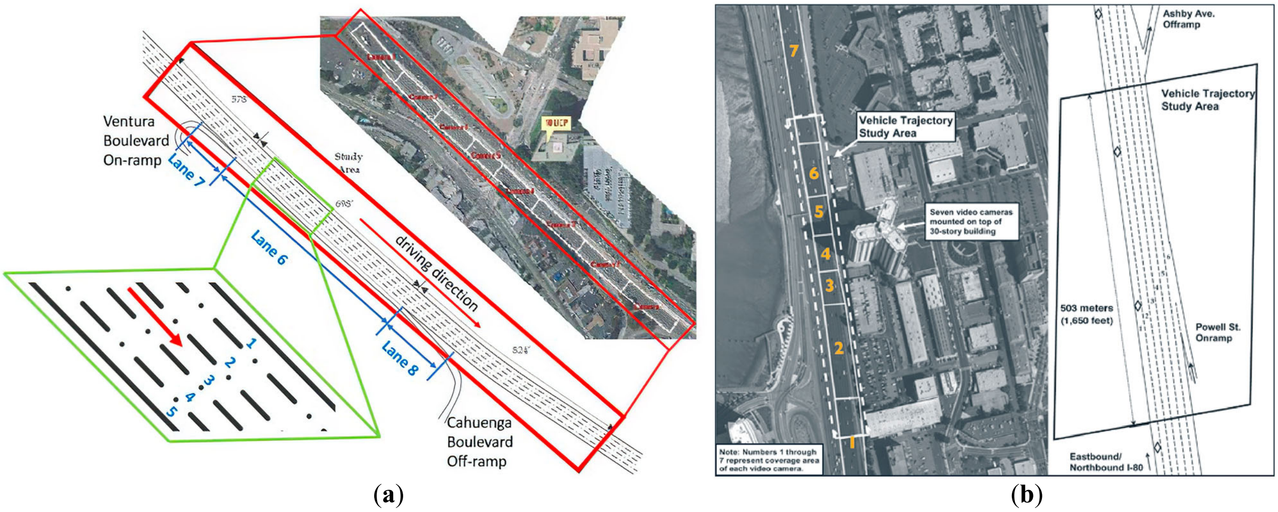
Type of maneuvre	Number of samples	Number of randomly sampled samples
lane keeping	1085089	34946
left lane change	134294	34946
right lane change	34946	34946

Considering the significant difference in the number of different maneuvres, in this experiment, the lane keeping and left lane changing data were randomly sampled to make their sample sizes consistent with that of right lane changing. Finally, the data are divided into the training set, testing set and validation set according to the ratio of 6:2:2. The number of each maneuvre is shown in Table 2.

In order to increase the training speed, alleviate overfitting, and reduce the model complexity, the data for model training were regularized. First, data differencing was performed to smooth the data features and eliminate the influence of fluctuations. The computational formula is as follows:

$$\mathrm{x}_{dif} = x_t - x_{t-1}, \tag{1}$$

where, x_{dif} is the data after feature scaling, x_t is the data at time t, and x_{t-1} is the data at time t-1.



(a)

(b)

Figure 1. Schematic diagram of roads used in the NGSIM dataset. (**a**) Road diagram for US_101; (**b**) Road diagram for I-80.

Then, feature scaling is performed on the differenced data to reduce the influence of different vertical and horizontal magnitudes on the model, so as to narrow the initial scope of data and reduce the influence of initial data differences on model training. The computational formula is:

$$\widetilde{x} = \frac{x_{dif} - x_{ave}}{(x_{dif\_\max} - x_{dif\_\min})/2}, \tag{2}$$

where, $\widetilde{x}$ is the processed data, $x_{dif\_\max}$ and $x_{dif\_\min}$ respectively, the maximum and minimum values of the data after feature scaling.

3. Methodology

The sampling attention encoder-decoder structure, global attention mechanism, and adaptive scheduled sampling method are discussed in detail.

3.1. Sampling attention encoder-decoder

This paper proposes a sampling attention encoder-decoder model based on the GRU neural network layer to model the driving scenario of the target vehicle. The specific structure of the model is shown in Figure 2.

The model inputs the target vehicle's trajectory within 8 s, incorporating the global driving features $\tilde{X} = \{S, v, \text{a}, \Delta S, \Delta v, \Delta \text{a}\}$ discussed in Section 2.1. These are divided into two parts by the observation moment t set in this study: the history trajectory representation and the future trajectory representation depicted in Figure 2. The history trajectory representation covers the global driving features from [0, t], while the future trajectory representation spans [t, 79]. The history trajectory representation is input into the encoder's GRU layers for encoding, generating the global embedding and the hidden state representation. The global embedding features encompass information on vehicle interaction behaviours and lane constraints. The mixed global embedding features and the future trajectory representation are input into the adaptive scheduled sampling module, which probabilistically replaces the global embedding calculated by the encoder with the actual future trajectory representation. This module assists the model in learning actual driving and interaction behaviours during training. The global embedding features obtained from the adaptive planning sampling module are combined with the encoder's hidden state to calculate global attention to determine the relevance between features, highlighting important interaction information. The attention information is concatenated with the global embedding features and input into the decoder GRU for decoding, outputting the normalized representation of the prediction trajectory.

3.2. Encoder-decoder model based on GRU

This study employs an encoder-decoder architecture to process trajectory sequences. This method encodes sequences into fixed-length context vectors to mitigate these issues and better understand the global information of the entire historical trajectory. The encoder-decoder's performance can be influenced by altering the neural network layers within it. In the field of trajectory prediction, researchers commonly use LSTM or GRU to handle time series information. They encode the historical trajectory over a past period and then decode the encoded features to output the predicted trajectory. As illustrated, at time t, the network's input is x_t, and the updated hidden state from the previous moment is h_{t-1}.

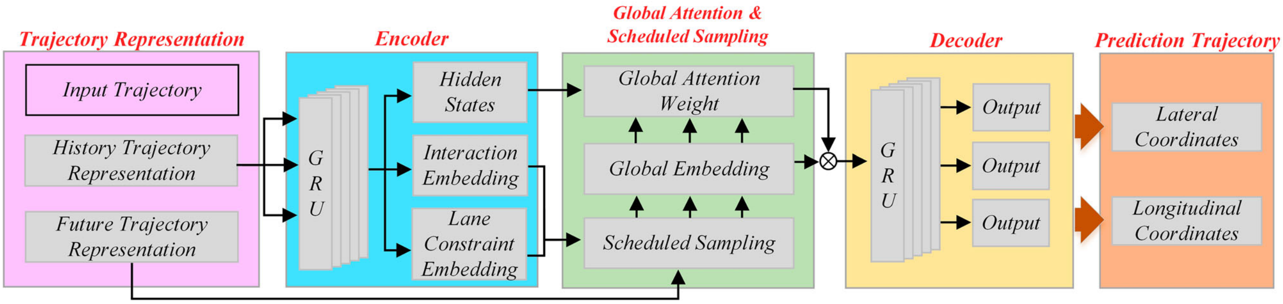


Figure 2. Structure of the prediction model (⊗ represents Concat).

Their computation processes in the reset gate r_t, update gate z_t, candidate hidden state $\tilde{h}_t$, and the final hidden state h_t are as follows:

$$r_t = sigmoid(W_r \cdot [h_{t-1}, x_t] + b_r), \quad (3)$$

$$z_t = sigmoid(W_z \cdot [h_{t-1}, x_t] + b_z), \quad (4)$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t] + b), \quad (5)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t. \quad (6)$$

Where, *sigmoid* represents the sigmoid activation function, and $*$ is the point-wise product. W_z, W_r, W, b_z, b_r, and b are model parameters. Within the encoder-decoder, the GRU serves as a neural network layer that encodes the input interaction features X into a compressed, fixed-length global feature encoding.

The driving behaviour of a vehicle at each moment is based on its driving decisions at some moment in the past. Therefore, the model must handle the observed vehicle motions and its interactive feature $\tilde{X} = \{S, v, \mathrm{a}, \Delta S, \Delta v, \Delta \mathrm{a}\}$ in the driving scene.

The structures of LSTM and GRU are shown in Figure 3. GRU can adaptively retain or forget past information according to the characteristics of the input sequence, which meets the requirements of long-term prediction tasks. Due to its lightweight characteristics, it can provide higher response speed and training speed.

The GRU model is used in this study. The input data is the global observation of the driving scene in the past, represented as $\tilde{X}^{\mathrm{t}} = \{S^{\mathrm{t}}, V^{\mathrm{t}}, \mathrm{a}^{\mathrm{t}}, \Delta S^{\mathrm{t}}, \Delta V^{\mathrm{t}}, \Delta \mathrm{a}^{\mathrm{t}}\}$, which includes interaction information about vehicles and lanes in the global driving scene. The lane constraints and vehicle interaction behaviours are encoded simultaneously through the GRU network layer in the encoding layer:

$$\tilde{X}_{embeding} = encoder(\tilde{X}^{\mathrm{t}}), \quad (7)$$

where, the *encoder* is the encoder, which contains 5 GRU layers, $\tilde{X}^{\mathrm{t}}$ is the actual driving features, and $\tilde{X}_{embeding}$ is the encoded features. The encoded features and the hidden layer are used as the output of the encoder part for global attention computation. The global encoded features encoded by the encoder, along with the encoder's current hidden state, are subjected to attention computation to assess their correlation. The global attention module aids the model in focusing on significant interaction features while disregarding less important ones. The specific principles and processes of global attention computation will be elaborately explained in Section 3.4. The decoder has the same structure as the encoder, with 5 GRU layers. After the global attention computation, the encoded features are decoded to obtain the predicted trajectories computed by the model.

3.3. Adaptive scheduled sampling module

Scheduled sampling is a training strategy commonly used with RNNs. In this study, we employ a scheduled sampling approach that probabilistically utilizes actual historical driving scene information as the target for global attention computation. Focusing on interactive behaviours in actual driving scenarios aids the decoder in learning correct driving behaviours. We adopt an adaptive ratio adjustment method to prevent the model from becoming overly reliant on actual driving scene data and enhance the overall generalizability of the prediction model. This approach dynamically reduces the frequency of model corrections using actual scene data based on the current training effectiveness of the model, thereby enhancing its self-optimization capabilities. The adaptive scheduled sampling is illustrated in Figure 4.

In Figure 4, global embedding refers to the global interaction feature encoding output by the encoder. Ground truth denotes the actual driving scenario information. Adaptive scheduled sampling probabilistically replaces the global encoded features obtained by the encoder with global interaction features from real scenarios at an adaptive rate. This approach assists the decoder in learning real driving behaviours during training, rather than just the features encoded by the encoder. After replacing the global encoded features with real driving scenario information, it is used together with

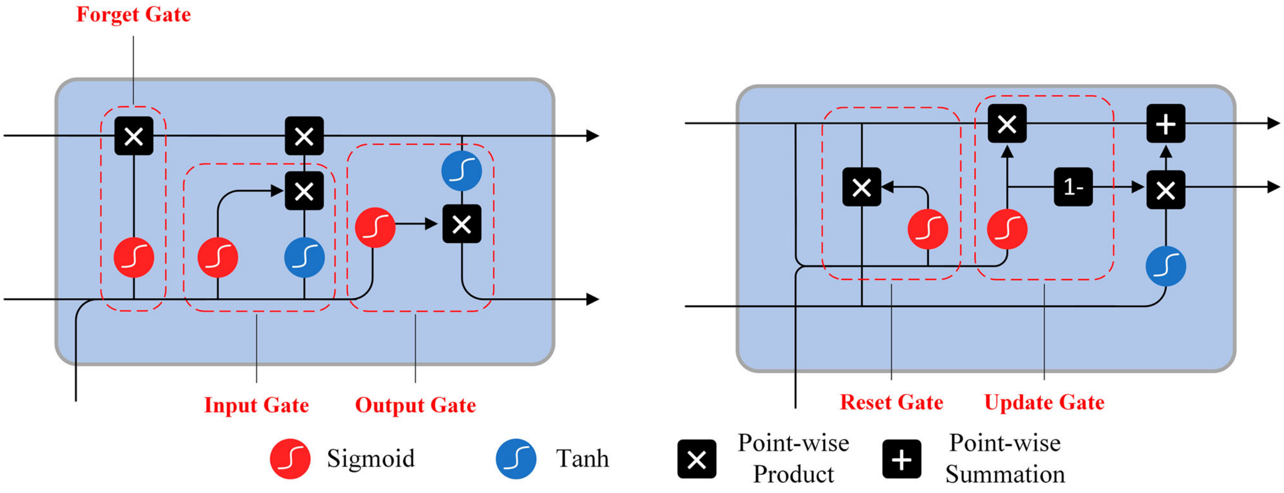


Figure 3. Structure diagram of LSTM and GRU.

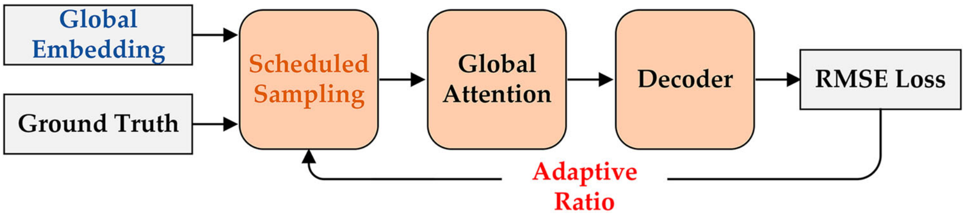


Figure 4. Schematic diagram of adaptive scheduled sampling.

the encoder's hidden state for the computation in the global attention model. The probability of scheduled sampling is adjusted based on the loss from the previous epoch. It probabilistically uses the feature information from real scenarios as the input for global attention calculation. Ultimately, the influence of real scenarios is wholly excluded, and only the output from the encoder is used as the input for the decoder. This ensures the model's ability to predict independently.

The scheduled sampling module utilizes a nonlinear hyperparameter design to assess the loss function from the model's previous iteration, determining the current predictive performance and adjusting the probability parameter for replacing the global embedding with ground truth. When the model's loss function falls within a specific range [a, b], the probability parameter α for this range is applied. In each iteration, the model generates a random number s between [0,1]; if s $< \alpha$, the ground truth replaces the global embedding. Intuitively, the lower α is, the more the model tends to rely on the global embedding obtained from the encoder for subsequent computational operations. Given that excessive reliance on real scenarios can diminish the model's generalization capabilities, the design of the scheduled sampling module sets the hyperparameter α to 0 once the model reaches a certain loss function threshold, thereby exclusively using the global embedding from the encoder for further calculations.

3.4. Global attention module

During the driving process, the driving characteristics of different vehicles at different moments contribute differently to the predicted trajectory of the target vehicle, and the information of all driving scenarios observed in the past cannot be computed equally (Vaswani et al., 2017). To address this issue, our work adopts the idea of the attention to extract the global interaction behaviours in the driving scene, compute the relationship between the encoded feature information and the hidden state, and focus on the essential parts of the scene sequence and the hidden state, to improve the expression and performance of the model. The schematic diagram of the global attention mechanism used in this paper is presented in Figure 5.

To extract the global attention weights in the past trajectories, the Hadamard product is performed on the global embedding $\tilde{X}_{embeding}$ and the last hidden layer H_L^T, so as to calculate the global attention weight matrix.*softmax* is used to process the global attention weights to reduce the influence of unconcerned parts on the model and highlight the critical information in the

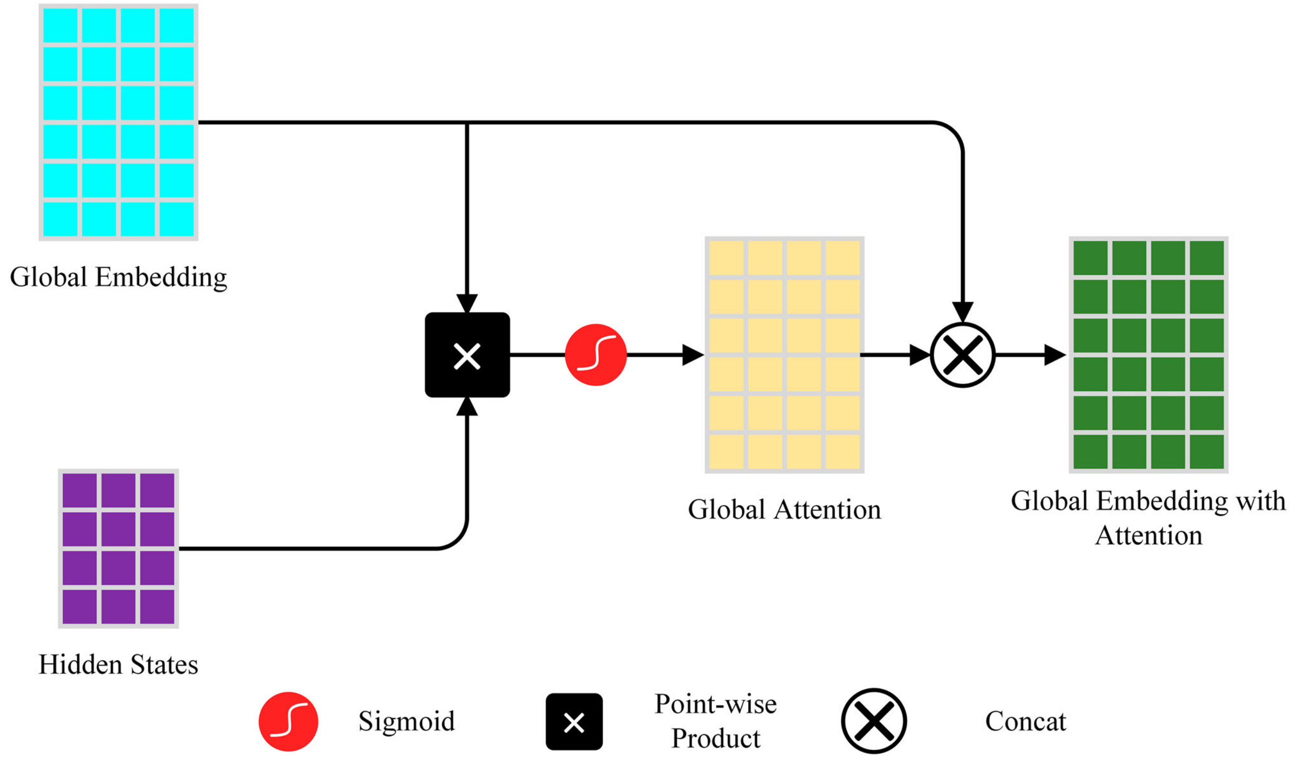


Figure 5. Calculation diagram of global attention matrix based on global coding and hiding state of driving scene at each moment.

scenario. The specific formula is as follows:

$$A = softmax(\tilde{X}_{embeding} H_L^T), \tag{8}$$

where, L represents the last layer of the hidden states; T denotes the transpose calculation of the matrix; *softmax* refers to that the *softmax* function is used as the activation function. The sequence weight A is obtained through the attention mechanism module, which concat with K to K_A, and the computational process can be represented as:

$$K_A = K \otimes A, \tag{9}$$

where, $\otimes$ denotes the concat operation. Before inputting the global attention matrix K_A into the decoder, in order to ensure that the encoded features are unified with the input dimension of the decoder, the feature information K_A obtained by splicing should be encoded through one linear layer to obtain the encoded feature matrix K':

$$K' = Linear(K_A). \tag{10}$$

Then, the encoded K' is input into the decoder for decoding, and after 5 decoding layers of GRU, the predicted trajectory (x^{t+1}, y^{t+1}) for the next moment is obtained. Based on that, the sliding computation is performed iteratively to obtain the predicted trajectory $\tilde{Y} = [(x^{t+1}, y^{t+1}), (x^{t+2}, y^{t+2}), \ldots, (x^{t+T}, y^{t+T})]$.

Table 3. Hyperparameters information table.

Parameter names	Parameters
Epochs	200
Batch size	1024
Optimizer	Adam
Learning rate	0.001
Weight decay	0.0001
Encoder-decoder network layers	5
Input time frames	30
Output time frames	50

4. Experimental results

4.1. Environment of the experiment

The experiments were conducted on a cloud server equipped with an NVIDIA TESLA P40 GPU. The training hyperparameters are shown in Table 3.

4.2. Loss function

The calculation method for the root mean square error (RMSE) is as follows:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(x_p(t) - x_t(t))^2 + (y_p(t) - y_t(t))^2}, \tag{11}$$

where, T represents the size of the prediction window or the number of time steps, which in this paper is 50. t

represents each step 1 to T, (x_t, y_t) represents the 2D coordinates of the actual trajectory, and (x_p, y_p) represents the two-dimensional coordinates of the predicted trajectory.

4.3. Evaluation metrics

Final displacement error (FDE): The displacement error per second for predicted trajectories versus real trajectories at 1 s, 2 s, 3 s, 4 s, and 5 s.

$$FDE = \sqrt{(x_p(t) - x_t(t))^2 + (y_p(t) - y_t(t))^2}, \qquad (12)$$

where, (x_t, y_t) represents the 2D coordinates of the actual trajectory, and (x_p, y_p) represents the two-dimensional coordinates of the predicted trajectory.

Average displacement error (ADE): The average error between the predicted trajectory and the real trajectory over 5 s, where T is 50 in this paper.

$$ADE = \frac{1}{T}\sum_{t=1}^{T}\sqrt{(x_p(t) - x_t(t))^2 + (y_p(t) - y_t(t))^2}, \qquad (13)$$

4.4. Comparison experiment

In order to evaluate the performance of the model in this paper in the extraction and prediction of interaction behaviours, the other methods are based on interaction behaviours and LSTM is used to extract temporal features from the data. Below is a list of the models used for the comparison experiments, along with a brief description of each model:

S-LSTM (Alahi et al., 2016): An LSTM encoder-decoder model that uses a social pooling layer to extract interactive behaviours.

CSP-LSTM (Deo & Trivedi, 2018): Builds on S-LSTM by adding a convolutional layer to extract interactive behaviours.

M-LSTM (Cui et al., 2019): An LSTM encoder-decoder model that integrates driving behaviour.

MATF-GAN (Zhao et al., 2019): A GAN encoder-decoder model with added convolutional layers.

ST-LSTM (Dai et al., 2019): This model embeds spatial interactions into the LSTM framework to measure interactions between neighbouring vehicles implicitly.

MHA-LSTM (Messaoud, Yahiaoui, et al., 2021): An LSTM encoder-decoder model that employs an attention mechanism to extract features of surrounding vehicles.

I-LSTM(SP) (Fei et al., 2019): An LSTM encoder-decoder model that integrates driving intentions, used on a 5 Hz sparsely sampled dataset.

SAED: The sampling attention encoder-decoder model proposed in this paper.

Table 4. Performance comparison of various trajectory prediction models.

	FDE of ever time step(m)				
Method	1 s	2 s	3 s	4 s	5 s
S-LSTM	0.65	1.31	2.16	3.25	4.55
CSP-LSTM	0.61	1.27	2.09	3.10	4.37
M-LSTM	0.58	1.26	2.12	3.24	4.66
MATF-GAN	0.66	1.34	2.08	2.97	4.13
ST-LSTM	0.54	1.16	1.88	2.70	3.63
MHA-LSTM	0.41	1.01	1.74	2.67	3.83
I-LSTM(SP)	**0.13**	**0.53**	**1.12**	2.18	3.43
SAED	0.38	0.83	1.29	**1.64**	**2.06**

The comparison results of our method with the above methods are shown in Table 4 and Figure 6.

The FDE of SAED is 0.38 m at 1 s, within 1 m at 2 s, and 1.5 m at 3 s, and the long-term prediction trajectory error is no more than 2 m at 4 s and 5 s. Considering that the lane width is usually 3 m, the proposed prediction model can generally ensure that the predicted trajectory and the actual trajectory are in the same lane, i.e. it can make accurate judgments of driving intentions (lane keeping, change to left lane and change to right lane). Meanwhile, when the prediction time is 4 s and 5 s, the I-LSTM error decreases by 24.77% and 39.94% compared with the second-highest accuracy.

4.5. Trajectory prediction consistency analysis

To visualize the stability of the model proposed in this paper, box plots were used to illustrate the results of ten experiments, as shown in Figure 7. The horizontal axis lists the Final Displacement Error (FDE) and Average Displacement Error (ADE) of the prediction model from 1 s to 5 s, while the vertical axis displays the error values in metres (m). The box includes error data ranging from the 25th to the 75th percentile. The median is represented by a solid red line, and the orange point indicates the position of the mean error. The whiskers extend to include data within 1.5 times the interquartile range (IQR). IQR is calculated as the value at the 75th percentile minus the value at the 25th percentile.

According to Figure 7, the errors at 1 s are all within 0.5 m, and those at 2 s are kept within 1 m. Meanwhile, the median of the errors at this prediction time length is low, suggesting that there are fewer predictions with high errors and the overall prediction errors are low. This indicates that SAED can adapt well to short-term predictions. At the prediction time of 3 s–5 s, the errors of the model increase, but the distribution of the prediction errors is still dense, and the prediction errors are within 3 m. This suggests that the model can accurately determine the driving intention. Still, the complexity of human driving behaviours leads to a dispersed distribution of the model prediction errors.

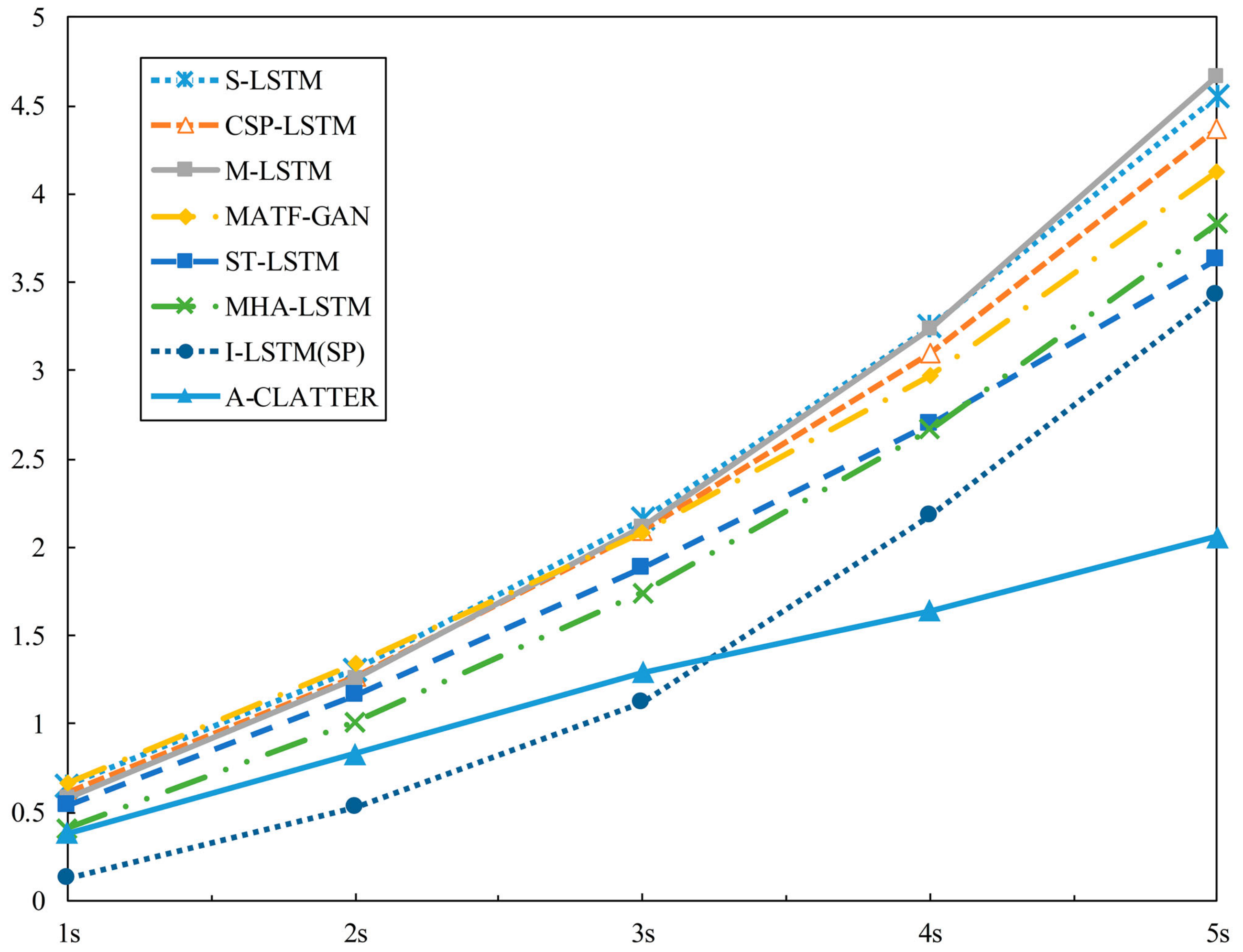


Figure 6. Performance comparison diagram of various trajectory prediction models.

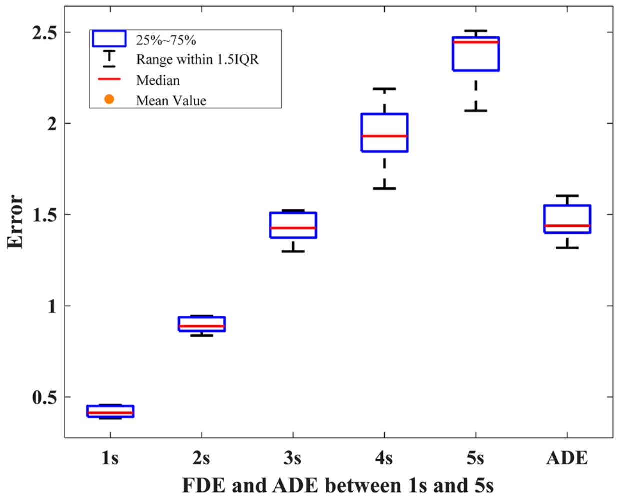


Figure 7. FDE and ADE box plots of trajectory prediction results.

4.6. Trajectory visualization

Figure 8 shows the 2D coordinate graphs of the predicted trajectories of the target vehicle obtained by our model and the actual trajectories, where (a), (b), and (c) show the right lane change, left lane change, and lane keeping driving behaviours, respectively. In each of these plots, the solid black lines represent the history trajectories, the red dotted lines represent the prediction trajectories in the future, and the blue dashed lines represent the ground truth in the future. It can be seen that when the prediction time is short, the prediction model has good performance, which is consistent with the actual motion trajectory of the target vehicle; when the prediction time is 4–5 s, the predicted trajectory may have some error, but it is still within the acceptable range, which does not affect the judgment of the travelling lane and driving intentions of the target vehicle.

4.7. Ablation study

4.7.1. Experimental design and results

In the ablation study, we compare the roles of different modules in trajectory prediction tasks by combining them in various configurations. The various models of

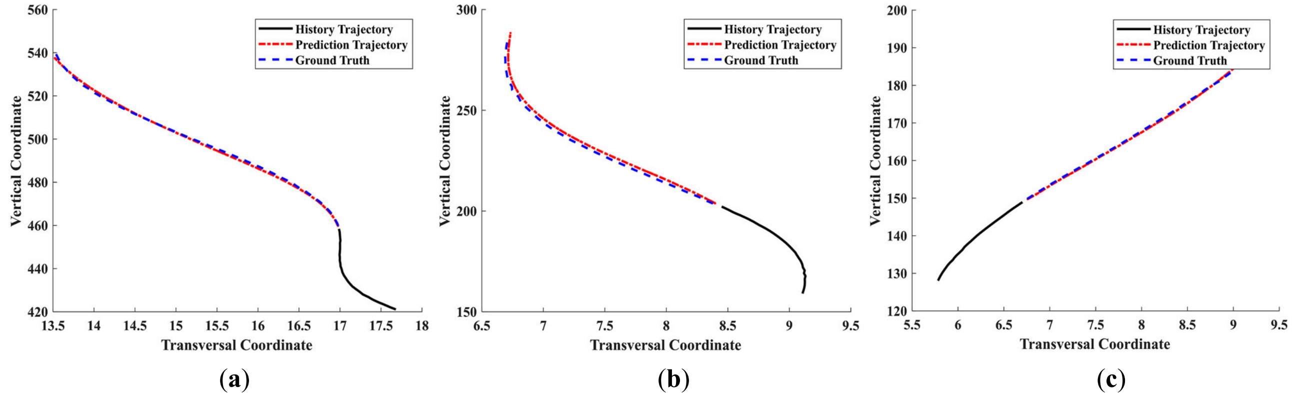


Figure 8. Comparison chart of predicted trajectories vs. actual trajectories under different driving intentions. (a) Right lane change; (b) Left lane change; (c) Lane keeping.

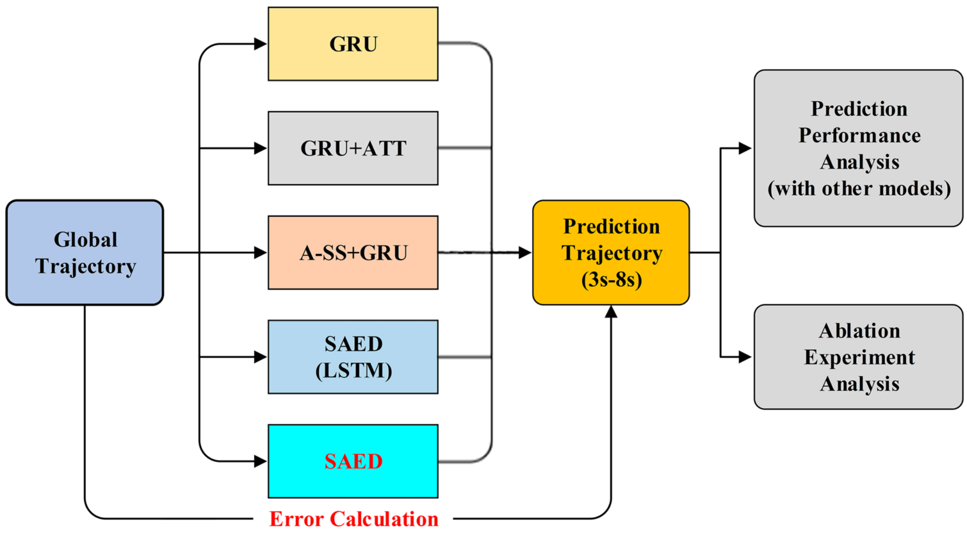

Global Trajectory
GRU
GRU+ATT
A-SS+GRU
SAED (LSTM)
SAED
Error Calculation
Prediction Trajectory (3s-8s)
Prediction Performance Analysis (with other models)
Ablation Experiment Analysis

Figure 9. The flowchart of the ablation experiment.

ablation experiments are described as follows, and the experimental procedure is illustrated in Figure 9.

GRU: GRU as the baseline encoder-decoder neural network layers.

GRU + ATT: GRU with an Attention mechanism for extracting interactive behaviours.

A-SS + GRU: Adaptive Scheduled Sampling with GRU for decoder correction.

SAED(LSTM): LSTM as the neural network layers in the encoder-decoder, establishing the same structure as SAED. It incorporates a global attention mechanism and an adaptive scheduled sampling model, employing the same hyperparameter settings.

Metrics include FDE, ADE, Epoch (number of epochs to achieve RMSE $<$ 3 over five iterations, indicating training time efficiency), and Storage Space (disk space used by the model, reflecting parameter size). The results of the ablation study are shown in Table 5. The FDE and ADE of the predicted trajectories for each model are illustrated in Figures 10 and 11.

4.7.2. Results analysis

For predictions over 3 s, the Sampling Attention module notably enhances model performance. The SAED model outperforms the Baseline by 30.65%, 35.43%, and 33.97% in accuracy at 3 s, 4 s, and 5 s, respectively, with a 31.28% improvement in ADE. GRU + ATT shows FDE improvements of 13.98%, 22.44%, and 8.97% at these intervals, with a 12.82% increase in ADE. A-SS + GRU also demonstrates better performance, with FDE increases of 16.13%, 21.65%, and 5.13%, and a 15.90% rise in ADE. These results suggest A-SS's effectiveness in learning actual driving behaviour, enhancing overall prediction. Additionally, ATT integration significantly boosts long-term accuracy by extracting interactive behaviours in dynamic scenarios. FDE and ADE comparisons across the five models are depicted in Figures 10 and 11.

Table 5. Comparison of the performance of different SAED components in ablation experiments.

	FDE(m)							
Model	1 s	2 s	3 s	4 s	5 s	ADE(m)	Epoch	Storage space (kb)
GRU	0.39	1.10	1.86	2.54	3.12	1.95	97	3683
GRU + AT	0.38	0.99	1.60	1.97	2.84	1.70	101	3675
A-SS + GRU	0.39	0.98	1.56	1.99	2.96	1.64	76	**3674**
SAED(LSTM)	0.43	0.95	1.30	1.81	2.69	1.46	**70**	4879
SAED	**0.38**	**0.83**	**1.29**	**1.64**	**2.06**	**1.34**	74	3675

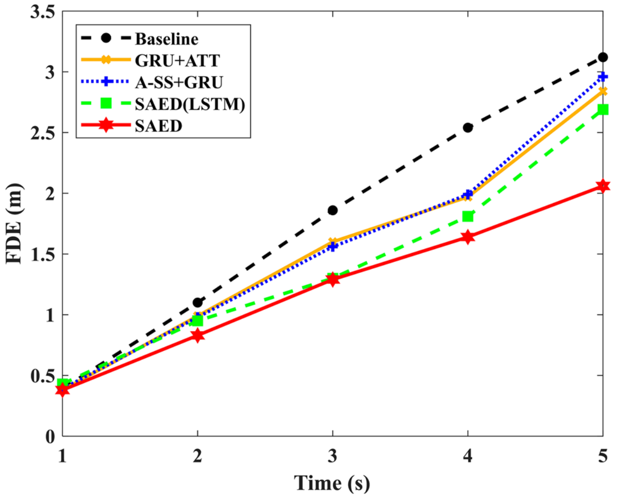


Figure 10. FDE of prediction trajectories of various model in ablation experiment.

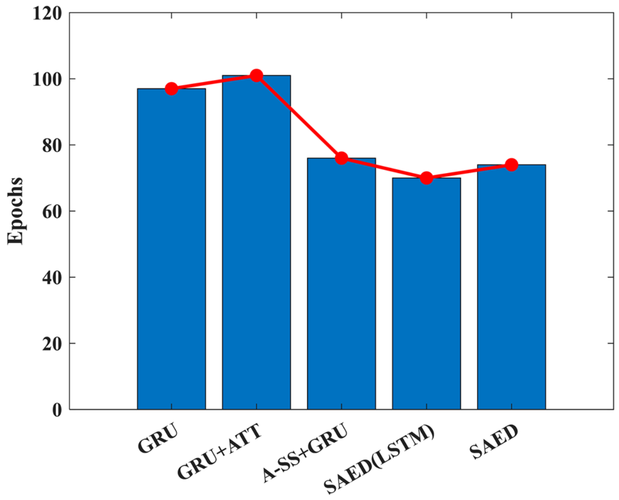


Figure 12. Histogram of training speed comparison of different models in SAED ablation experiment.

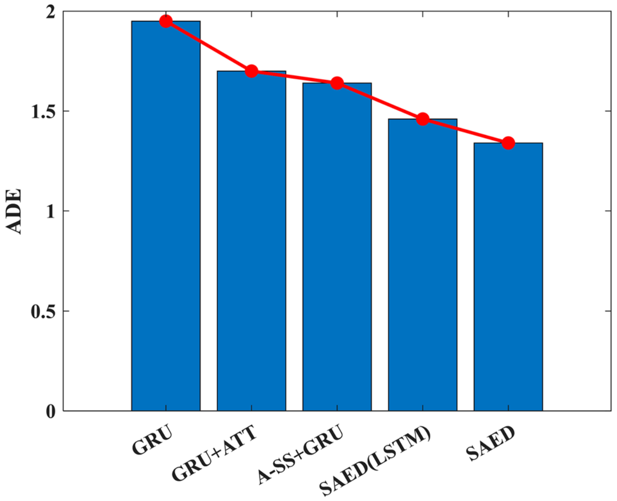


Figure 11. Histogram of ADE of prediction trajectories of various models in ablation experiment.

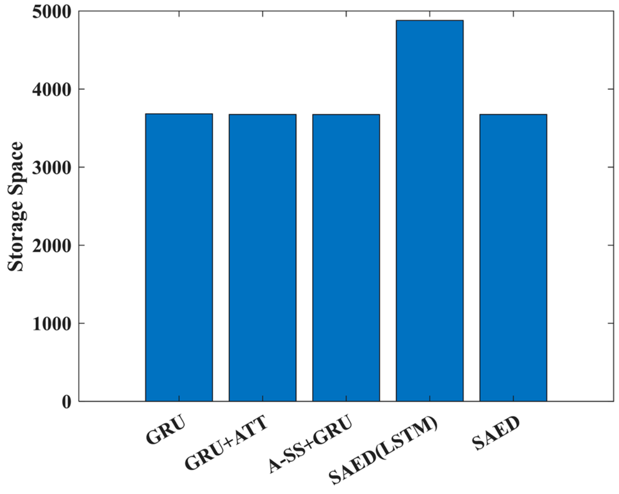


Figure 13. Histogram of storage space comparison of different models in SAED ablation experiment.

Furthermore, the A-SS module is also shown to improve the fitting speed of the model significantly. Compared with the baseline, A-SS reduces the Epoch required for the trajectory prediction model to reach the same performance by 21.65% and ADE by 15.90%. The comparison of the training speed of each model is shown in Figure 12.

Compared with SAED(LSTM), the SAED can reduce the model occupancy by 24.68% without affecting the performance, significantly reducing the computational resources and memory space required by the model. A comparison of the storage space occupied by each model is shown in Figure 13.

5. Conclusions

This paper builds a deep learning method named SAED for predicting the trajectories of target vehicles in interactive scenarios. Initially, a global attention model is established to extract interaction features from the global embedding of interactive scenes, enhancing the model's learning ability of interactive behaviours and long-term predictive performance. An adaptive scheduled sampling model probabilistically utilizes actual driving scenarios to train the trajectory prediction model to address the challenges of the model's slow learning of actual driving behaviours and the lack of initial correction in encoded features during training. This approach increases global attention to actual interactions and reduces dependency on actual scenes as the model's training improves, thereby enhancing the generalizability of the prediction model. Lastly, GRU is chosen as the neural network layer for the encoder-decoder to minimize the model's parameter size and computational dependency. SAED achieves an ADE of 1.34 m over 1–5 s on the NGSIM dataset.

Additionally, its FDE for 4 s and 5 s trajectory predictions are 1.64 and 2.06 m. Compared to the commonly used LSTM method, the model's storage space is reduced by 24.68%. This demonstrates its effectiveness in extracting potential interactive behaviours in driving scenarios with limited computational resources, thereby improving the model's long-term predictive capability.

Author contributions

Conception and design, Z.Y., Y.W. and L.D.; analysis and interpretation of the data, Z.Y., Y.W., X.Y. and Y.H.; drafting of the paper, Y.W. and W.Z.; revising article critically for intellectual content, Z.Y., Y.W., L.D. and X.Y.; the final approval of the version to be published Z.Y. and Y.H. All authors agree to be accountable for all aspects of the work.

Data availability statement

The data that support the findings of this study are openly available in U.S. DOT Intelligent Transportation Systems (ITS) Public Data Hub at http://doi.org/10.21949/1504477, reference number 14.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the Natural Science Foundation Project of Chongqing Municipal Science and Technology Bureau (No. cstc2020jcyj-msxmX0803), the Chongqing Key Laboratory Foundation of Manufacturing Equipment Mechanism Design and Control (No. KFJJ2018056) and the Scientific Research Foundation of Chongqing Technology and Business University (No. 1956056 and No. 2152003).

References

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., & Savarese, S. (2016, June). Social LSTM: Human trajectory prediction in crowded spaces. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 961–971). IEEE.

Altché, F., & de La Fortelle, A. (2017, October). An LSTM network for highway trajectory prediction. In *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 16-19 October 2017 (pp. 353–359).

Cui, H., Radosavljevic, V., Chou, F.C., Lin, T.H., Nguyen, T., Huang, T.K., Schneider, J., & Djuric, N. (2019, May). Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)* (pp. 2090–2096).

Dai, S., Li, L., & Li, Z. (2019). Modeling vehicle interactions via modified LSTM models for trajectory prediction. *IEEE Access*, *7*, 38287–38296. https://doi.org/10.1109/ACCESS.2019.2907000

Deo, N., & Trivedi, M. M.. (2018). *Convolutional Social Pooling for Vehicle Trajectory Prediction. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 1468–1476).

Fei, C., Ji, X., He, X., Yulong, L., & Liu, Y. (2019). Intention recognition and trajectory prediction for vehicles using LSTM network. *Zhongguo Gonglu Xuebao/China Journal of Highway and Transport*, *32*. https://doi.org/10.19721/j.cnki.1001-7372.2019.06.003

Gindele, T., Brechtel, S., & Dillmann, R. (2010, September). A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems*, 19-22 September 2010 (pp. 1625–1631).

Kim, B., Kang, C. M., Kim, J., Lee, S. H., Chung, C. C., & Choi, J. W. (2017, October). Probabilistic vehicle trajectory prediction over occupancy grid Map via recurrent neural network. In *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 16-19 October 2017 (pp. 399–404).

Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H. S., & Chandraker, M. (2017). *DESIRE: Distant future prediction in dynamic scenes with interacting agents* (pp. 336–345).

Lefèvre, S., Laugier, C., & Ibañez-Guzmán, J. (2013). *Intention-aware risk estimation for general traffic situations, and application to intersection safety*. Report. INRIA.

Mercat, J., Gilles, T., El Zoghby, N., Sandou, G., Beauvois, D., & Gil, G. P. (2020, May). Multi-head attention for multi-modal joint vehicle motion forecasting. In *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*, 31 May 2020 - 31 August 2020 (pp. 9638–9644).

Messaoud, K., Deo, N., Trivedi, M. M., & Nashashibi, F. (2021, July). Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation. In *Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV)*, 11-17 July 2021 (pp. 165–170).

Messaoud, K., Yahiaoui, I., Verroust-Blondet, A., & Nashashibi, F. (2021). Attention based vehicle trajectory prediction. *IEEE Transactions on Intelligent Vehicles*, *6*(1), 175–185. https://doi.org/10.1109/TIV.2020.2991952

Tang, C., & Salakhutdinov, R. R. (2019). Multiple futures prediction. In *Proceedings of the advances in neural information processing systems* (Vol. 32). Curran Associates, Inc.

U.S. Department of Transportation - FHWA. (2022). *The next generation simulation (NGSIM).* Retrieved January 15, 2024, from https://doi.org/10.21949/1504477

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the advances in neural information processing systems* (Vol. 30). Curran Associates, Inc.

Yamak, P. T., Yujian, L., & Gadosey, P. K. (2020, February 7). A comparison between ARIMA, LSTM, and GRU for time series forecasting. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, 20-22 December 2019 (pp.49–55). Association for Computing Machinery, New York, NY, USA.

Yang, S., Yu, X., & Zhou, Y. (2020, June). LSTM and GRU neural network performance comparison study: Taking yelp review dataset as an example. In *Proceedings of the 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, 12-14 June 2020 (pp. 98–101).

Zhang, L., Zhang, J., Niu, J., Wu, Q. M. J., & Li, G. (2021). Track prediction for HF radar vessels submerged in strong clutter based on MSCNN fusion with GRU-AM and AR model. *Remote Sensing*, *13*(11), 2164. https://doi.org/10.3390/rs13112164

Zhang, Y., Shi, X., Zhang, S., & Abraham, A. (2022). A XGBoost-Based lane change prediction on time series data using feature engineering for autopilot vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *23*(10), 19187–19200. https://doi.org/10.1109/TITS.2022.3170628

Zhao, T., Xu, Y., Monfort, M., Choi, W., Baker, C., Zhao, Y., Wang, Y., & Wu, Y. N. (2019). *Multi-Agent Tensor Fusion for Contextual Trajectory Prediction. In 2019 International Conference on Robotics and Automation (ICRA)* (pp. 12126-12134).