Environment-Attention Network for Vehicle Trajectory Prediction

Yingfeng Cai, *Senior Member, IEEE*, Zihao Wang, Hai Wang, *Senior Member, IEEE*, Long Chen, Yicheng Li, Miguel Angel Sotelo, *Fellow, IEEE*, and Zhixiong Li, *Senior Member, IEEE*

***Abstract*—In vehicle trajectory prediction, the difficulty in modeling the interaction relationship between vehicles lies in constructing the interaction structure between the vehicles in the traffic scene. Majority of existing models only focus on the interaction between the historical trajectory of the vehicle and the surrounding vehicles in the spatial domain, and do not pay attention to the interaction between the vehicle and the non-Euclidean correlation structure (graph structure) that exists in the environment. In order to overcome the deficiencies in the existing models, this paper proposes the Environment-Attention Network model (EA-Net) to obtain the full interactive information between the vehicle and its driving environment. In the proposed model, a new type of parallel structure consisting of Graph Attention network (GAT) and Convolutional social pooling containing Squeeze-and-Extraction mechanism (SE-CS), is constructed as the environmental feature extraction module and embedded in LSTM encoder-decoder. This structure solves the limitation of the dimension and structure influence when constructing the interaction model between the vehicle and the surrounding environment, making the extracted feature information comprehensive and effective. The prediction accuracy of the model with the RMSE loss function is tested on two public datasets– NGSIM and highD, and compared with several state-of-the-art trajectory prediction algorithm models. The results show that the prediction accuracy of the proposed Environment-Attention Network in the two datasets is more than 20% higher than that of the single structure model, which indicates that the proposed model proposed has superior performance and better adaptability to different traffic environments compared with the existing models.**

***Index Terms*—Intelligent vehicle, trajectory prediction, environment-attention network, graph attention network, squeeze-and-extraction mechanism.**

Manuscript received May 6, 2021; revised August 1, 2021 and August 20, 2021; accepted September 7, 2021. Date of publication September 9, 2021; date of current version November 18, 2021. This work was supported in part by the National Natural Science Foundation of China under Grants U20A20333, 52072160, 51875255, and U1764264, in part by the Natural Science Foundation of Jiangsu Province under Grants BK20180100 and BK20190853, in part by Key Research and Development Program of Jiangsu Province under Grants BE2019010-2, BE2020083-2, and BE2020083-3, in part by Jiangsu Province's six talent peaks under Grant TD-GDZB-022, and in part by Australia ARC DECRA under Grant DE190100931. The review of this article was coordinated by Prof. Zhanyu Ma. *(Corresponding author: Hai Wang.)*

Yingfeng Cai, Long Chen, and Yicheng Li are with the Automotive Engineering Research Institute of Jiangsu University, Zhenjiang 212013, China (e-mail: caicaixiao0304@126.com; chenlong@ujs.edu.cn; liyucheng070@163.com).

Zihao Wang and Hai Wang are with the School of Automotive and Traffic Engineering of Jiangsu University, Zhenjiang 212013, China (e-mail: 756574958@qq.com; wanghai1019@163.com).

Miguel Angel Sotelo is with the Department of Computer Engineering, University of Alcal, 28801 Alcal de Henares, Spain (e-mail: miguel.sotelo@uah.es).

Zhixiong Li is with the Yonsei Frontier Lab, Yonsei University, Seoul 03722, Republic of Korea (e-mail: zhixiong.li@yonsei.ac.kr).

Digital Object Identifier 10.1109/TVT.2021.3111227

I. Introduction

In recent years, as an emerging field that is constantly developing, smart cars are providing more convenient and effective services to the society. With the advancement of smart car technology, the smart systems such as vehicle collision avoidance systems and driver assistance systems provide good assistance to the drivers. The advanced intelligent systems enable the drivers and the passengers to drive the vehicles in a safer and more comfortable traffic environment.

The various systems contained in a smart car need the support of a large amount of surrounding environment information during operation. Sine a smart car cannot fully reach the driving level of a human driver and will interact with surrounding vehicles to varying degrees while the vehicle is driving on the road. Therefore, the vehicle needs to make reasonable path planning [1], [2], [34], [36] according to the future trajectory of itself and the surrounding vehicles. When the driver overtakes, changes lanes or exhibits any other behaviors in different traffic scenarios, he needs to consider the driving environment of the vehicle, i.e., the position, relative distance and relative speed contained in the surrounding vehicles. Therefore, providing the system with more accurate future trajectories of vehicles in dynamic traffic scenarios under complex conditions is a priority issue. While evaluating the vehicle trajectory prediction system, not only the accuracy of the prediction should be considered, but also the generalization ability of the established model, that is, whether the model accurately can predict the trajectory of the vehicle in different traffic scenarios. Recently, the interactive perception relationship between the vehicle and its surrounding environment has become a hot research topic. The existing research has proposed many different vehicle trajectory prediction models and methods, aiming to construct the interactive perception relationship between the vehicles [3]–[6], [35].

This paper proposes a novel trajectory prediction model and a modeling method of vehicle interaction in dynamic traffic scene. First of all, we improve the vehicle interaction model based on occupancy grid graph, and extends the correlation between vehicles to the level of the combination of spatial structure and graph structure. In particular, in order to better integrate the two structures, we propose a graph structure relationship based on occupancy grid graph to fit the interaction relationship between vehicles. At the same time, in order to extract the interaction between the two levels of vehicles, we built a neural network model, Environment Attention Network (EA-Net), which can

increase the attention mechanism of various elements in the Environment according to the different structures of interaction. The proposed model is horizontally expanded on the basis of the structure of Long Short-Term Memory (LSTM) encoder-decoder and Convolutional social pooling [3] in series. We improve the Graph Attention network (GAT) [23], and combine the GAT with the Convolutional social pooling containing the Squeeze-and-Excitation Block (SE) [24] to form a parallel structure. The GAT enhances the model's learning of the interactive behavior of each vehicle in the graph structure of the dynamic environment composed of vehicles. In addition, we use an adaptive adjacency matrix to initialize the edge relationship between vehicle nodes. When the model extracts the temporal features in space, different attention weights are assigned to the temporal channel information of different spatial locations. The use of attention mechanism suppresses the participation of useless information in time when the model parameters are updated, and makes the key information more effective, which significantly improves the efficiency of the model at runtime.

This new type of parallel structure is used to capture the feature information updated by each node in the graph structure formed by the vehicle and the surrounding environment during the driving process, as well as the feature information in the spatial location structure of the surrounding environment. Compared with the Convolutional social pooling model, the new proposed model structure has significant improvement in the effect of extracting the environmental interaction information, and at the same time achieves a better trajectory prediction effect than other existing models.

The contributions of this research are summarized as follows:

- This paper establishes an interactive relationship model between vehicles in dynamic traffic scene, which integrates occupancy grid graph and graph structure. The interaction between vehicles not only retains the spatial dependence between vehicles, but also constructs the hidden graph structure interaction between vehicles, which makes the interaction between vehicles in the same scene more closely related to the multi-dimensional features contained in each vehicle.
- We propose a new model structure, which includes a novel environment feature extraction module. This module consists of an improved Graph Attention network and Convolution social pooling containing SE module. We introduce the attention mechanism into the environment feature extraction module, so that it can extract and integrate the interactive information between the spatial structure and the graph structure more efficiently and accurately.

The rest of the article is arranged as follows: Section II provides an introduction to the related work of trajectory prediction research. Section III discusses the current research issues and presents detailed definitions. The structure of the proposed model is described in Section IV. In Section V, the experiment contents and the numerical results are discussed. Finally, the conclusions are provided in Section VI.

II. Related Works

The existing vehicle trajectory prediction methods can be divided into two categories: non-interactive models based on vehicle maneuver, and interactive perception models between vehicles based on Deep Learning.

The model based on vehicle maneuver first recognizes the vehicle maneuver. This step can be attributed to the classification problem based on the characteristics of the vehicle [7]. For example, classifiers based on heuristics [8], Bayesian networks [9], hidden Markov models [10], random forest classifiers [11] and Gaussian process models [12], such models are input to vehicle historical trajectories. Firstly, the kinematics parameters are determined by the classifier to determine the vehicle maneuver. Then the regression trajectory prediction model is used to predict the future trajectory of the vehicle. There are also trajectory prediction models that use shallow neural networks. Yoon *et al.* [13] used a multi-layer perceptron (MLP) for lateral motion prediction. They proposed three representative trajectories for each lane according to the speed of the vehicle reaching each target lane. The MLP model provides the probability that the vehicle will select each lane and have each possible trajectory. With the development of Deep Learning, the role of recurrent neural network (RNN) [14] for trajectory prediction has also been discovered. Numerous trajectory prediction models have joined the variant LSTM of RNN for trajectory prediction [15]. The use of LSTM can better solve the problem of long-term dependence.

However, these models only consider the various factors of the vehicle itself, but do not include the influence of the surrounding vehicles or even the surrounding environment on the future driving trajectory of the vehicle. In addition, in real-time driving scenarios, different traffic scenes and driving styles of drivers will increase the complexity of vehicle maneuvers. The vehicle maneuver type recognition algorithm predefined by the model cannot accurately identify the correct maneuver in such situations. Incorrect maneuver recognition will significantly increase the error of trajectory prediction.

Recently, Deep Learning has caused a significant impact in various research fields [30], [31], [33]. Deep Learning has shown excellent effect in dealing with regression problems. Interactive perception model based on Deep Learning is the mainstream direction of recent research on trajectory prediction. This type of model can directly perform the end-to-end modeling of the prediction task, and also considers the influence of the interaction between the observed vehicle and the surrounding vehicles on the future trajectory of the vehicle. Deo *et al.* used the LSTM encoder to encode the trajectory vectors of surrounding vehicles and added vehicle maneuvers to predict the future trajectory of the vehicle [16], [17]. In the trajectory prediction models proposed in [18], [19], the vehicles surrounding the observing vehicle in the same traffic scene are constructed into an occupancy grid map. In [18], a new LSTM encoder-decoder model has been proposed that uses the occupancy grid map and the beam search algorithm to predict the trajectory of the vehicle. The beam search algorithm optimizes the original greedy algorithm of the LSTM mechanism, calculates the multiple

possible trajectories at the same time, and reserves the best trajectory for each decoding iteration. Deo *et al.* [3] embedded the LSTM-encoded vehicle trajectory vector into the corresponding position in the occupancy grid to construct a convolutional social pool tensor, which could be used to represent the historical trajectory of surrounding vehicles in space and the observed vehicle interaction. In addition, they also regarded the future trajectory of the vehicle as a probability distribution based on the multi-peak vehicle movement and used the model to predict the parameter values of the probability distribution. Messaoud *et al.* [4] optimized the Convolutional social pooling in [3], and proposed a non-local social pooling that increased the attention mechanism. GAIL-GRU [20] used a generative adversarial network to imitate the randomness of the future trajectory when a human was driving a vehicle, and achieved the purpose of predicting the future trajectory by learning to imitate the driving strategy of the driver. The Multi-Agent Tensor Fusion Model [5] utilizes its full convolution module (U-Net) to summarize the LSTM encoding vector of each vehicle in the traffic scene and the CNN-encoded vehicle scene graph, and uses the Generative Adversarial Network (GAN) for the future motion trajectories of multiple agents. The trajectory prediction model proposed by Jeon *et al.* [6] and Huang *et al.* [21] regarded the relationship network formed by all agents in the same scene as a graph structure expressed by non-Euclidean distances, and used graph neural networks to obtain the interactive features between each node in the graph structure. Although, all the above-mentioned models consider the environmental interaction features within a certain structure, the extraction of environmental features always only considers a single interactive structure. Thus, the extracted environmental interaction features are not completely sufficient. In the model proposed in this paper, this problem is well optimized and the prediction accuracy of the vehicle's future trajectory is improved.

In recent years, many studies have used datasets collected for specific scenarios and specific traffic environments for experiments, but the model trained with such datasets has the problem of insufficient generalization ability. Some studies also take into account the limitation of road on vehicle movement and mark the road in the dataset to obtain vehicle positioning [21]. For comparison with most existing studies, two large public datasets, NGSIM [26], [27] and highD [28], were used to test and verify our proposed method. These two datasets contain different traffic rules, road geometry, regional traffic flow differences between two countries, which can cover most driving conditions well and ensure the accuracy of the test and the generalization ability of the model.

The RMSE function is used to evaluate the final results in most of the current studies. The function calculates the offset between the predicted trajectory coordinates and the real coordinates at each time point in the predicted trajectory, and presents it visually in the form of distance. We also used this function to evaluate the results of each experiment and compare them with existing studies to ensure the fairness of the comparison experiment.

III. Problem Formulation

The interaction between vehicles in the same traffic scene and their surrounding vehicles is modeled. For any vehicle in the traffic scene at each time t, the vehicle will interact with the surrounding vehicles at the spatial location level. This spatial location structure is constructed using the occupancy grid map. In addition, the characteristic information between the vehicles will also be transmitted and updated in a non-Euclidean distance structure (graph structure). Therefore, this graph structure of information transmission between vehicles is constructed through nodes and connections.

A. Initialize Input Features

In a static traffic scene at a certain moment, the basis for observing any behavior that the vehicle is about to perform comes from two levels: The first is based on the current time point. The characteristic values (vehicle position, vehicle speed, acceleration, heading angle, and relative distance) in various states of the historical trajectory of the observed vehicle will affect the future behavior of the vehicle and the generation of the future trajectory; The second is the interaction between the various states of the surrounding vehicles in the historical trajectory of the observed vehicle and the state of the observed vehicle, including the influence of the historical spatial position of the surrounding vehicles on the observed vehicle and the interaction between the characteristics of the historical trajectory of the surrounding vehicles and the observed vehicle.

Analyzing from the perspective of the driver operating the vehicle at a certain moment: When the driver drives the vehicle, he evaluates the current driving environment through the positions, angles and relative speeds of the surrounding vehicles and the driving vehicle. For different driving environments, the driver will decide which action the vehicle should perform at the current moment based on the driving experience to change the state of the vehicle at the next moment. This enables the vehicle to adapt to the various impacts of surrounding vehicles on the subsequent safe driving of the vehicle in the current driving environment.

In this paper, the state feature $x_i^{(t_{obs})}$ of the observed vehicle at time *t* is used as the benchmark to initialize the state features contained in the historical trajectories of all vehicles in the traffic scene at the current time. Then, the state feature x_k at a certain moment in the historical trajectory of the kth vehicle (including the observed vehicle *i*) in the current traffic scene will be initialized to $\tilde{x}_k$:

$$\tilde{x}_k^{(t_{obs}-t_i)} = x_k^{(t_{obs}-t_i)} - x_k^{(t_{obs})} \tag{1}$$

where X_k is the historical trajectory feature sequence of the vehicle and is expressed as:

$$X_k = \left\{\tilde{x}_k^{(t_{obs}-T)}, \ldots, \tilde{x}_k^{(t_{obs}-1)}, \tilde{x}_k^{(t_{obs})}\right\} \tag{2}$$

$$\tilde{x}_k = \{x, y, v, a, \theta, d\} \tag{3}$$

Among them, $t_i \in T, k \in n$, n is the number of all vehicles in the current traffic scene, and T is the time length of the historical

trajectory of the vehicle. The characteristics of each moment are the above-mentioned relativized horizontal and vertical coordinates (x, y), vehicle speed v, acceleration a, heading angle θ, and the relative distance d.

B. Occupancy Grid Map Construction

The predicted vehicle lane and the vehicles traveling in the two surrounding lanes are used to construct an occupancy grid map centered on the observed vehicle. The width W_g and the length L_g of each grid in the occupancy grid map are equal to the width of the lane $(W_g = W_{lane})$, and the length of a standard car $(L_g = L_{veh})$, respectively. The vehicles within the range of the grid map are placed into each grid in the occupancy grid map according to their specific positions to form the spatial feature information map of the vehicles around the observed vehicle in the traffic scene. Moreover, the time series features of each vehicle are embedded into the corresponding position in the occupancy grid map to construct the space-time feature tensor between the vehicles.

C. Graph Construction

The graph structure proposed in this paper is constructed based on the above-mentioned occupancy grid map. When a vehicle is driving on a road, it will form a topological graph structure that will expand along the road with the information existing in each location node in the traffic scene where the vehicle is located (the proposed graph structure is described in detail in Section 4).

The graph structures proposed in [6], [21] all establish the inner agent in the same scene as a fully connected graph structure in which each node is connected, that is, the default initial adjacency matrix has the same edge value of 1 between each node. However, the interactive relationship (connecting edges) between the vehicle and the vehicle in the image may change during the driving of the vehicle. Therefore, another method of graph construction is adopted in this paper. The interaction between the observed vehicle and the surrounding vehicles is converted into the interaction between the observed vehicle and the various components in the surrounding environment. This definition method is similar to the modeling method in [22] for predicting the traffic speed in the graph structure constructed using the speed sensor data at fixed points in the city.

This static relationship is applied to the dynamic environment around the vehicle, that is, each graph structure consisting of the surrounding environment and the vehicles at every moment is regarded as a static structure, and ignore its essence that it is a dynamic structure. The information contained in all the grids in the occupancy grid map is regarded as features of the surrounding environment around the observed vehicle and each grid is regarded as a node V in the surrounding environment.

The environment node will interact with the observed vehicle regardless of whether there are vehicles within the road range represented by the environment node.

As shown in Fig. 1, the environment node V_j in the occupancy grid diagram $(W_g \times L_g)$ around any observed vehicle V_i on the road is defined as the first-order neighbor of the vehicle node.

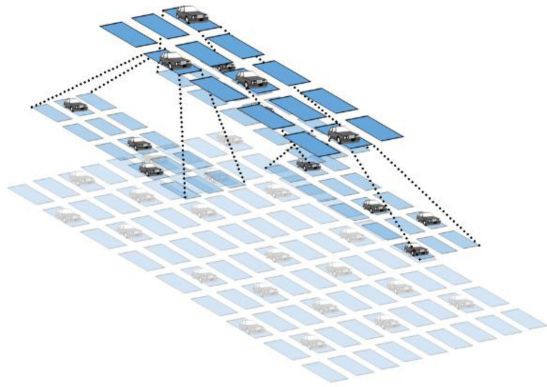


Fig. 1. Each vehicle on the road forms a graph structure with the surrounding environment nodes. The first layer is the first-order neighbor nodes around the vehicle, having a direct connection between them. The kth level is the kth order neighbor nodes of the vehicle that have no edges between the vehicles but will transmit information through the (k-1) th order neighbors.

The distance between each node is 1, $(d(V_i, V_j) = 1)$. We define that the observed vehicle will be affected by these first-order neighbor nodes during the driving process, so that the change of driving state will occur, namely the updating of features. When there is a vehicle in the grid, the node feature H is the vehicle temporal information $h^{(t)}$ in the grid. If there is no vehicle in the node, the node information is represented as an all-zero vector of the same dimension, meaning that the value of the environmental feature at the node location is 0, $(H = 0)$.

Since the area occupied by the grid diagram is less than the area of the overall road, if there are vehicles in the environment node V_i in the occupying grid diagram, when the vehicles inside the node are the observed vehicles, the surrounding environment nodes will also interact with V_{j} , and these nodes will also act as the high-order neighbor nodes of V_i.

D. Loss Function

The root mean square error (RMSE) loss function is widely used to evaluate the accuracy of model prediction [3]–[6]. The RMSE loss function can well reflect the average distance error between the predicted and the observed trajectory values during the prediction time. The calculation formula for RMSE loss function is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left((x_{i,t} - \hat{x}_{i,t})^2 + (y_{i,t} - \hat{y}_{i,t})^2\right)} \quad (4)$$

where n is the total number of vehicles tested, $x_{(i,t)}$, $y_{(i,t)}$ are the position coordinates of the i-th vehicle at predicted time t, and $\hat{x}_{i,t}$, $\hat{y}_{i,t}$ are the observed values of the position coordinates.

IV. Proposed Model

In this section, the specific architecture of the proposed Environment-Attention Network (EA-Net) model is described in detail. Fig. 2 shows the overall structure of the network. The network consists of a feature extraction module and a trajectory prediction module. The feature extraction module includes a

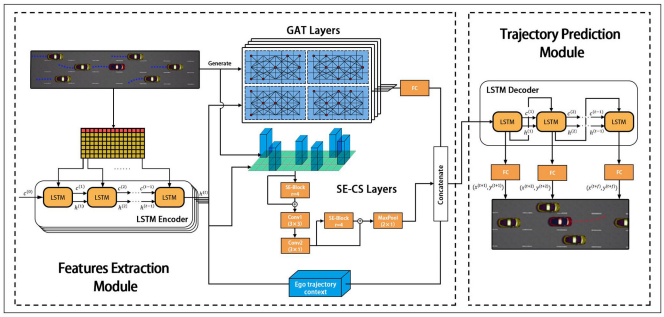


Fig. 2. The overall structure of the proposed Environment-Attention Network (EA-Net) model consisting of a feature extraction module and a trajectory prediction module. The feature extraction module includes an LSTM encoder and an environmental feature extraction module composed of Graph Attention Network (GAT) and SE Convolutional Social Pool (SE-CS), which outputs a context vector containing all feature information. The trajectory prediction module includes an LSTM decoder that decodes the context vector input by the feature extraction module and outputs a predicted future trajectory.

time feature encoder and an environmental feature extraction module. The time feature encoder is composed of an LSTM encoder, and its role is to encode the historical trajectory features of the observed vehicle and all the vehicles in the surrounding grid. The environmental feature extraction module is composed of GAT [23] and a Convolutional social pool containing SE module [24] to extract the interactive feature information in the graph structure and the spatial location structure of the surrounding environment of the vehicle. This module outputs the context vector composed of time series, spatial and graph features of the observed vehicle and its environment. The trajectory prediction module is composed of an LSTM decoder that receives and decodes the context vector and finally outputs the predicted future trajectory.

A. LSTM Encoder

The LSTM network is used to extract the temporal feature information, and each LSTM cell shares the weights. The input of the encoder is the historical trajectory feature X_i of the observed vehicle and the surrounding vehicles in the same traffic scene in formula (2). The hidden state update in the LSTM cell at time t is expressed as:

$$c_i^{(t)} = c_i^{(t-1)} \cdot g_f\left(h_i^{(t-1)}, f_e(X_i^{(t)})\right) + g_{in}\left(h_i^{(t-1)}, f_e(X_i^{(t)})\right) \tag{5}$$

$$h_i^{(t)} = tanh\left(c_i^{(t)} \cdot g_{out}\left(h_i^{(t-1)}, f_e(X_i^t)\right)\right) \tag{6}$$

where f_e is the embedding function, g_f, g_{in}, and g_{out} are the forget gate, the input gate, and the output gate inside the LSTM, respectively, c_i^l is the cell memory of the i-th vehicle at time t, and $h_i^{(t)}$ is its hidden state at time *t*.

B. Environmental Feature Extraction Module

1) Graph Feature Extraction Network: Each grid (regardless of whether there are vehicles inside) in the occupancy grid graph built around the observed vehicle is regarded as an environment node. Then, the constantly moving vehicle and the surrounding environment nodes form a dynamic fixed node graph structure.

The collection of each node and edge in the graph is expressed as $G = (V, E)$, where V represents the set of nodes in the graph, E represents the set of edges existing between each node, and F is the feature dimension of feature H_k contained in node V_i. Thus, the feature of each node in the surrounding environment of the observed vehicle can be expressed as:

$$H_k = \begin{cases} h_j \ if \ \ k = j \\ 0 \quad else \end{cases} \tag{7}$$

The number of environment nodes is equal to the total number of grids in the grid graph N, $i \in N$. h_j is the temporal feature extracted by the LSTM encoder of the j-th vehicle in the occupancy grid, $H_k \in R^F$.

The interaction of nodes in the environment graph structure is an undirected graph structure. The changes in the characteristics of each node will cause the feature update of its neighbor nodes, which will eventually drive the feature changes of all nodes in the entire graph structure. This change is considered as a kind of transmission and diffusion of graph signals. Each node will receive the characteristic information passed to it by the surrounding nodes through the edges, and at the same time, will pass its own characteristic information to the surrounding nodes. An improved multi-layer GAT is used to extract the graph interaction features between each node in the graph.

The Graph Attention layer builds a graph attention weight matrix containing the attention between nodes by learning the attention weight values between the nodes. Compared with the Laplacian matrix L in the graph convolution layer, the weight matrix strengthens the weight coefficient of the edges between the nodes. Taking node V_i as the central node, the attention weight coefficient e_{ij} of V_i and a certain neighbor node V_j is expressed as:

$$e_{ij} = a\left(W_v H_i, W_v H_j\right) \tag{8}$$

Where W_v is the feature update weight of each node in the attention layer of the current graph, and a is the attention function between two nodes. A convolutional layer with a convolution kernel size of 1 is used as a fully connected layer to aggregate the feature information between the two nodes and a scalar representing the correlation between the nodes is calculated. Then, the activation function is used to form the attention coefficient e_{ij}:

$$e_{ij} = LeakyReLU\left(W_a * \left([W_v H_i || W_v H_j]\right)\right) \tag{9}$$

where $\|$ is the concatenation operator, and $W_a \in R^{2F \times 1}$. In order to better distribute the weights, the coefficients of the attention mechanism of the *SoftMax* function are used for row normalization to obtain the attention weight matrix α:

$$\alpha_{ij} = \frac{\exp\left(LeakyReLU\left(W_a * \left([W_v H_i || W_v H_j]\right)\right)\right)}{\sum_{k \in V_i} \exp\left(LeakyReLU\left(W_a * \left([W_v H_i \ || W_v H_k]\right)\right)\right)} \tag{10}$$

Fig. 3. shows the feature update process of each environment node according to the attention weight matrix in the graph attention layer. In order to avoid the inaccurate connection

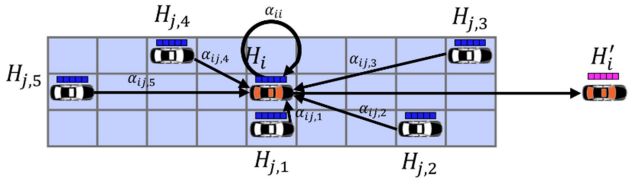


Fig. 3. Feature update process of each environment node in the graph attention layer. The blue vector is the historical trajectory feature contained in the environment node where the vehicle exists. The node where the orange vehicle is located updates its feature H_i to $H_i^{'}$ according to the attention coefficient α and the feature H_j of each neighbor node of the existing vehicle.

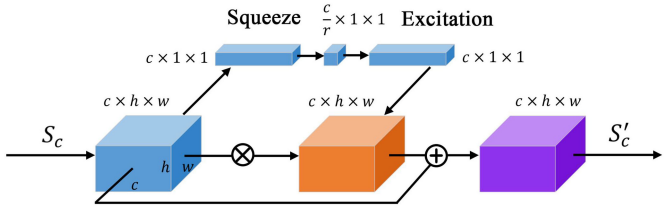


Fig. 4. The upper path of the Squeeze-and-Excitation Block (SE-Block) input tensor S_c squeeze and excitation operations that aggregate the channel features through global average pooling and reduce the size of S_c to 1×1. After squeeze and excitation, the attention weight is obtained, multiplied by S_c, and then added to S_c to obtain the social tensor S_c' with additional attention features.

between the environment nodes defined by human subjectively, an adaptive adjacency matrix A_f is constructed as the initial adjacency matrix input to the Graph Attention layer:

$$A_f = SoftMax\left(LeakyReLU\left(M_1 M_2^T\right)\right) \tag{11}$$

where$M_1, M_2 \in R^{N\times F}$ are the two learnable parameter matrices. After the two parameter matrices are subjected to matrix multiplication, a parameter matrix $M, M \in R^{N\times N}$ is obtained. This parameter matrix is normalized by row as the adjacency matrix input to the attention layer in the proposed model. The model uses Leaky ReLU as the activation function for the nonlinearization of the feature matrix to obtain better results than ReLU. The use of Leaky ReLU retains the negative value points in the parameter matrix. Hence, the negative value of the edge becomes a relatively small positive value after the *SoftMax* line normalization, avoiding too many unlearned neurons. The edge strength parameters in the adjacency matrix A_f will be shared among the attention layers. In the process of model training, the adjacency matrix A_f establishes the edges between the nodes in the graph through continuous learning and updating. At the same time, A_f and the attention weight matrix α can be modified after learning and updating, which enhances the accuracy of the expression of the strength of the edges between the nodes.

The obtained attention weight matrix is used to update the features of each node in each layer of the graph, and the convolution operation is used to compress the feature parameters of the output of the attention layer of the L-layer graph to compress the channel feature number, $W_o \in R^{\sum^F \times F}$. The final extracted graph feature H_G formula calculation can be expressed as:

$$H^l = A_f\left(\sum_{i\in N}\alpha_{ij} W_g H^{l-1}\right) \tag{12}$$

$$H_G = W_o{}^*\left(||_{l=1}^{L} H_l\right) \tag{13}$$

2) Spatial Feature Extraction Network: The GAT described above extracts the interactive features existing in the graph structure formed between the vehicle and the surrounding environment nodes. However, when the vehicle is actually running, the spatial position information of the surrounding vehicles relative to the observed vehicle is also crucial. The Convolutional social pooling [3] is improved and used to extract the interactive relationship features in the spatial position of the historical trajectory sequence of the observed vehicle and the surrounding vehicles in the traffic scene.

Firstly, a three-dimensional social tensor S_c is constructed whose two spatial dimensions are the same as the size of the occupancy grid map, and number of channels is equal to the hidden state dimension output by the LSTM encoder. Then the Squeeze-and-Excitation Block [24] (SE-Block) is used as the spatial attention mechanism in the time channel of the constructed space-time tensor. The role of SE-Block is to better model the spatial dependence. In the spatial feature extraction network proposed in this paper, the SE-Block is used twice for feature enhancement. As shown in Fig. 4, the social tensor S_c is first input into the adaptive average pooling layer, whose role is to squeeze the spatial size ($h\times w$) of the input social tensor S_c. The purpose of this step is to encode the spatial information between the channels into a global feature and embed that feature in all channels in order to ensure that the global feature is shared by all channels. The global feature $Z_c \in R^c$ is calculated as:

$$Z_c = F_{sq}\left(S_c\right) = \frac{1}{h\times w}\sum_{i=1}^{h}\sum_{j=1}^{w} S_c\left(i,j\right) \tag{14}$$

An excitation mechanism is used for Z_c to obtain the correlation between the channels. The essence of the excitation mechanism is a bottleneck layer structure composed of a fully connected layer (FC) that consists two convolutional layers with a convolution kernel size of 1. This structure effectively increases the generalization degree of the model. *r* is the channel dimension reduction coefficient, which is a hyperparameter in this module. The first FC layer reduces the channel dimension of Z_c to *c*/*r*, and the second layer FC restores the channel dimension to *c*. After this step, the activation function σ is input and finally the attention weight tensor *s* is output. The calculation formula is expressed as:

$$s = F_{ex}\left(W, Z_c\right) = \sigma\left(g\left(W, Z_c\right)\right) = \sigma\left(W_2 * ReLU\left(W_1 * Z_c\right)\right) \tag{15}$$

where$W_1 \in R^{\frac{c}{r}\times c}, W_2 \in R^{c\times\frac{c}{r}}$ are the parameter matrices. The attention weight *s* is expanded to the same spatial size as the social tensor S_c, denoted as s_e. Then, operations on S_c are performed to obtain the social tensor $\tilde{S}_c$ containing the channel attention, and its formula is:

$$\tilde{S}_c = F\left(S_c, s_e\right) = S_c * s_e \tag{16}$$

where F is the channel multiplication operation between the social and the attention tensors S_c and s_e, respectively, while $*$ is the Hadamard product symbol.

In addition, a structure similar to that in the ResNet [25] is used to add $\tilde{S}_c$ and the initial social tensor S_c to obtain a new social tensor $S'_c, (S'_c = \tilde{S}_c + S_c)$. The addition of $\tilde{S}_c$ enables the social tensor to retain the original features, and the attention mechanism is used to enhance the characteristic information in the channel based on the spatial correlation of each channel. Among them, S_c, as a kind of identity mapping, can also effectively prevent the problem of affecting the gradient calculation when the gradient disappears in the SE-Block.

After the two-layer convolution operation, the SE-Block is used again to enhance the channel spatial correlation feature, and finally the spatial feature H_S between the observed vehicle and the surrounding vehicle time series is output through the maximum pooling layer.

C. Trajectory Prediction Module

The trajectory prediction module is composed of LSTM decoder and fully connected network. The function of trajectory prediction module is to decode the context vector extracted in the above-mentioned module to predict the coordinates of the vehicle's future trajectory. The compressed graph feature H_G is passed through the fully connected layer and then dimensionality reduction processing is performed, turning H_G into a one-dimensional feature vector as a context vector that can represent the graph structure composed of all environment nodes. Similarly, dimensionality reduction is performed on H_S and the hidden state of the observed vehicle. Finally, the temporal characteristics of each spatial location are summarized in the form of context vectors. The characteristics of the input to the LSTM decoder can be expressed as:

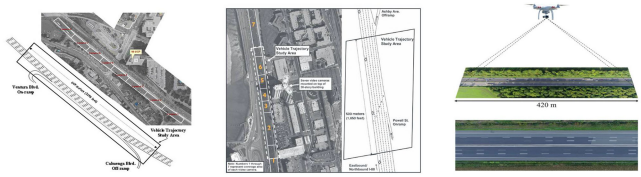
$$H_{context} = g\left(H_G, H_S, h_i^{(t)}\right) \tag{17}$$

where g is the fusion function of three feature context vectors. In this paper, a concatenation method is used to fuse the three-dimensional features extracted from the proposed model with the vehicle maneuvering state. The resulting context vector $H_{context}$ contains the interactive features in time, space and graph, and summarizes the hidden features of the observed vehicle in various dimensions.

The LSTM decoder receives and decodes the $H_{context}$, and outputs the hidden state vector $H_{pred} = \{h_{pred}^{(t+1)}, h_{pred}^{(t+2)}, \ldots, h_{pred}^{(t+f)}\}$. Then, the future trajectory coordinates are output through the fully connected layer:

$$(x^{(t)}, y^{(t)}) = f\left(h_{pred}^{(t-1)}, W_{FC}\right) \tag{18}$$

where $(x^{(t)}, y^{(t)})$ are the vehicle trajectory coordinates at time t, and W_{FC} is the parameter matrix of the fully connected layer.



NGSIM US-101 [28] NGSIM I-80 [27] highD [29]

Fig. 5. Datasets used in the experiment: The data in the NGSIM datasets was collected by multiple digital video cameras. The US-101 highway [27] includes five mainline lanes and an auxiliary lane connecting the on-ramp, while the I-80 highway [26] is a one-way six-lane highway. The highD dataset [28] was collected by drones, measuring the road length of 420m, and the data of the two-way six-lane sampling points is selected for testing.

V. Experiment

A. Dataset and Model Training

Two public datasets are used to evaluate the prediction accuracy of the proposed model. Sample images from the two datasets are shown in Fig. 5. The first dataset is the public dataset in the Next Generation Simulation (NGSIM) [26], [27] research project initiated by the US Federal Highway Administration. The NGSIM dataset contains the data of all vehicles passing through the US-101 [27] and the I-80 [26] highways in a total of 45 minutes. The dataset corresponding to each highway is divided into three 15-minute segments under different traffic conditions. The dataset comes from the image information extracted by digital video cameras set up on the highway, and the camera samples at a frequency of 10 Hz. The dataset provides the relative vehicle coordinates on the two highways. The lengths of US-101 and I-80 highways are about 640 and 503 meters, respectively. In addition, each piece of data also contains information such as vehicle speed, acceleration, vehicle type, lane, and front and rear vehicle IDs.

The second dataset is the highD dataset [28], which is a large-scale natural vehicle trajectory dataset of German highways collected by drones. The sampling frequency of the dataset is 25 Hz, and the dataset includes 16.5 hours of measured driving data of 110000 vehicles in 6 locations. The total driving distance of the vehicles is 45000 kilometers, and a total of 5600 complete lane changes have been recorded.

From the NGSIM dataset, 70% of the data in all time periods is selected as the training set, 10% of the dataset is selected as the validation set during the training, and the remaining 20% of the data is used as the test set. Due to the large amount of data in the highD dataset and the two different road scenes including the two-way four-lane and the two-way six-lane, the two-way six-lane vehicle trajectory data is selected as the experimental data. The data is divided into training, validation and testing sets the same as the NGSIM dataset. The 3 seconds historical trajectory of the vehicle is used to predict the trajectory coordinates of the vehicle in the next 5 seconds. The NGSIM and the highD datasets selected in the experiment contain 7940071 8 seconds and 12710320 8 seconds trajectory data, respectively. Each input trajectory is processed and 5 Hz is used as the sampling frequency to select the new trajectory sampling points

in the trajectory to make the input trajectory coordinate curve smoother. Among them, each trajectory point in the historical trajectory contains the horizontal and the vertical coordinates of the vehicle, vehicle speed, acceleration, and heading angle.

The proposed model is trained using the Adam optimizer [29], the batch size is 128, The weight of attention output from each GAT layer is fed into the dropout layer, the dropout ratio is 0.2, and the negative slope of the activation function Leaky ReLU is $\alpha = 0.1$. The optimizer uses an exponentially decreasing learning rate for training. The initial learning rate is 0.0005. During the training process, if the value of the verification set loss function exceeds the previous minimum value three times, the learning rate is multiplied by 0.7. The hidden state dimension of the LSTM encoder is 64, the output dimension of the LSTM decoder is 128, and the channel dimension of the input and output nodes of the GAT layer is 64. In addition, since the average speed of most vehicles in the NGSIM and the highD datasets falls between 20 km/h-60 km/h and 80 km/h/-140 km/h, respectively. Therefore, the size of the occupancy grid map contained in the model trained is defined using the NGSIM dataset to be (13×3) [3], that is, the number of environment nodes is 39, and the size of the occupancy grid map contained in the model trained using the highD dataset is defined as (41×3) [4], and the number of environment nodes is 123. The construction of the model is implemented using the PyTorch framework.

TABLE I
THE COMPARISON OF THE ROOT MEAN SQUARE PREDICTION ERROR (RMSE) BETWEEN THE PROPOSED MODEL AND THE EXISTING ADVANCED MODELS IN THE 5-SECOND PREDICTION RANGE ON THE NGSIM AND THE HIGHD DATASETS

Datasets	Prediction Horizons (s)	M-LSTM	CS-LSTM	MATF	NLS-LSTM	Scale-Net	EA-Net
NGSIM	1	0.58	0.61	0.67	0.56	0.46	0.42
	2	1.26	1.27	1.34	1.22	1.16	0.88
	3	2.12	2.09	2.08	2.02	1.97	1.43
	4	3.24	3.10	2.97	3.03	2.91	2.15
	5	4.66	4.37	4.13	4.30	-	3.07
highD	1	-	0.22	-	0.20	-	0.15
	2	-	0.61	-	0.57	-	0.26
	3	-	1.24	-	1.14	-	0.43
	4	-	2.10	-	1.90	-	0.78
	5	-	3.27	-	2.91	-	1.32

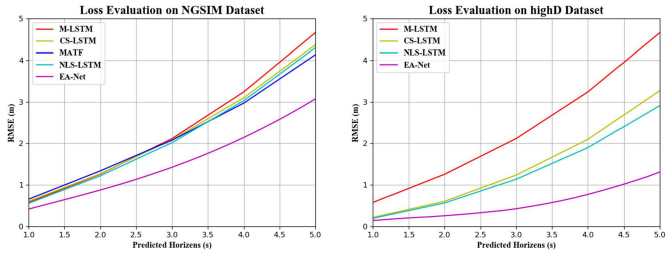


Fig. 6. The RMSE loss function curves of different models on NGSIM and highD datasets.

B. Model Prediction Effect Analysis

1) Model Prediction Accuracy Comparison: The prediction accuracy of the EA-Net model proposed in this article is compared with the following state-of-the-art models the NGSIM and the highD datasets.

- Maneuver-LSTM (M-LSTM) [16]:

The M-LSTM is an encoder-decoder-based model, where the encoder encodes the trajectory of the target and surrounding vehicles. The code vector and the mobile code are input to the decoder, and the decoder decodes them to generate multi-modal trajectory prediction.

- Convolutional Social Pooling (CS-LSTM)[3]:

The CS-LSTM is a model based on LSTM encoder-decoder and Convolutional social pooling. This model takes the lateral and the longitudinal maneuvers of the vehicle as input, for which a maneuver prediction module is also constructed. The output of CS-LSTM is a binary Gaussian distribution parameter.

- Non-local Social Pooling (NLS-LSTM) [4]:

The NLS-LSTM model establishes the Non-local social pooling that combines the local and the non-local information. The non-local multi-head attention mechanism captures the importance of each vehicle relative to the observed vehicle.

- Multi-Agent Tensor Fusion (MATF) [5]:

This model fuses the encoded single-agent tensor into a multi-agent tensor and embeds it into the scene tensor, then uses the U-Net to extract the features and input them into the GAN network to capture and predict the uncertainty trajectory of the agent's actions.

- Scale-Net[6]:

The Scale-Net is an agent trajectory prediction model based on LSTM encoder-decoder and Edge-enhance Graph Convolutional Neural Network. The predicted vehicle trajectory is output by the MPL.

Table I and Fig. 6 compare the results of the proposed model with the models proposed in recent studies under the RMSE loss function as the evaluation index. Compared with the Scale-Net model that also uses graph neural network, the prediction effect of the proposed model on the NGSIM dataset has an accuracy improvement of about 22%. Compared with other NLS-LSTM models that use convolutional social pools to extract the social features, the average improvement effect of the proposed model is greater than 27%. Compared with the existing research, the EA-Net has a significant improvement in the trajectory prediction effect of the last 3 seconds on the NGSIM dataset. Compared with the NGSIM dataset, the vehicle trajectory data extracted by the highD dataset has a significantly large improvement in accuracy. The improvement of the prediction effect of the proposed model on the highD dataset is more than 50%, which significant compared with the other models. When the proposed model is used to predict the vehicle trajectory in the highD dataset, the size of the occupancy grid map is expanded to (41,3). Therefore, the size of the occupancy grid map is increased. Hence, the amount of information contained in the map structure and the spatial location structure received by the environment extraction module in the model also increases, making the predicted trajectory more accurate. In addition, the reliability of the dataset is also one of the important reasons for the improvement of prediction accuracy.

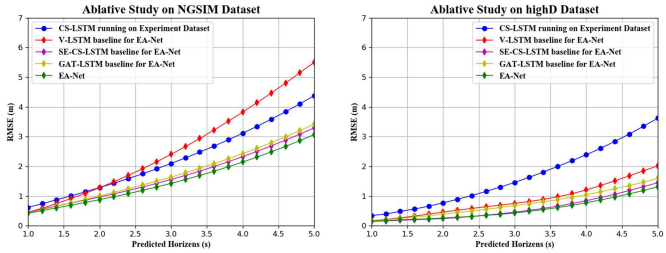


Fig. 7. Comparison of prediction effects of EA-Net's sub-models V-LSTM, SE-CS-LSTM, GAT-LSTM and the whole EA-Net under the RMSE loss function, where CS-LSTM is the baseline for comparison.

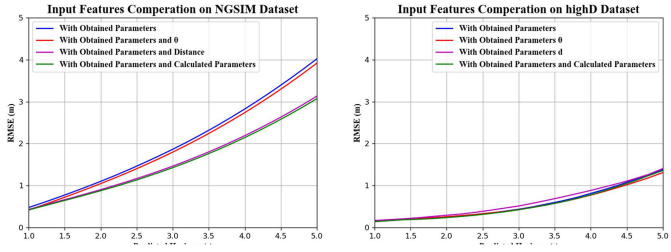


Fig. 8. The RMSE loss function curves of the model predicted trajectories under different initial inputs x_k.

2) Ablation Experiment: The proposed EA-Net model is decomposed into modules and the additive effect of the attention module in the model on the prediction accuracy is demonstrated through comparative experiments. Each sub-model in the experiment contains the LSTM encoder-decoder, and all models are trained with the same parameters setting. The separated sub-models include a vanilla-LSTM model, a model with an additional SE-CS module based on the LSTM encoder-decoder, a model with an additional GAT module based on the LSTM encoder-decoder, and a complete EA-Net model. At the same time, the CS-LSTM model is used as a baseline for comparison that is trained on the experimental dataset used in this paper, and outputs the predicted 5 seconds trajectory at a frequency of 5 Hz. As shown in Fig. 7, on the same NGSIM experimental dataset, the prediction accuracy of the CS-LSTM model is only higher than that of the V-LSTM model in the proposed model. The overall proposed model and the sub-model SE-CS-LSTM and GAT-LSTM have significantly improved the accuracy of predicting trajectories. The environmental feature extraction modules composed of GAT and SE-CS have played their respective roles. Compared with the sub-model that only contains one module, the EA-Net has a better extraction effect on environmental features. In the highD experimental dataset, the prediction effect of the fully connected LSTM codec model in the proposed model also reaches a good level. This is because the initial input features of the proposed model provide a more comprehensive historical trajectory after being encoded. The information enables the decoder to predict a more accurate future trajectory based on historical trajectory information. In addition, because we adjusted the area occupied by the grid map according to the highD dataset, the prediction accuracy of SE-CS-LSTM model on the highD dataset is higher than that of GAT-LSTM model. In the next experiment, the effects of models with different initial inputs on trajectory prediction are compared.

3) Research on Input Features: Aiming at the influence of the initial input feature x_k of the model proposed in the ablation experiment on the prediction accuracy, the prediction effect of the model is tested with different inputs x_k. First, the input features x_k of the observed vehicle and the surrounding vehicles in the model are divided into obtained and calculated parameters. The obtained parameters include vehicle position coordinates (x, y), vehicle speed v, and acceleration a. The calculated parameters include the relative heading angle θ and the relative distance d between the observed vehicle and the surrounding vehicles. Different parameter combinations are input in the model and tested on two datasets. The obtained the results are shown in Fig. 8.

In the NGSIM dataset, the model that only inputs the obtained parameters and the model that inputs both the obtained parameters and the relative heading angle θ have a closer prediction effect, which indicates that the input of the relative heading angle θ has a weak effect on the prediction accuracy of the model. When the relative distance d is added to the input parameters, the prediction effect of the model significantly improve, which shows that the relative distance between the observed vehicle and the surrounding vehicles is an important factor affecting the future trajectory of the vehicle.

In the highD dataset, the combination of different input parameters has miniature effect on the accuracy of trajectory prediction. Among them, when the proposed model inputs three directly obtainable features (vehicle position, vehicle speed, and acceleration) and directly obtainable features plus the relative heading angle θ, the accuracy of the trajectory predicted by the model in 2-4s is slightly improved. However, the input of the relative distance d between vehicles does not significantly improve the prediction accuracy of the model. Therefore, it is believed that when the proposed model is run on the highD dataset, adding too many inputs cannot achieve better prediction results, and inputting appropriate initial feature parameters $x_k = \{x, y, v, a, \theta\}$ can better improve the accuracy of vehicle future trajectory prediction.

4) Impact of Maneuver on Prediction: The predicted vehicle's lateral direction is divided into three situations: lane keeping (LK), left lane change (CL), and right lane change (CR), and tested the prediction accuracy of the model under different lateral maneuvers. Fig. 9 shows the histograms tested under the two datasets. It can be seen from the figure that the proposed model has the best predictive effect on the trajectory of straight vehicles in the two datasets, that is, the average RMSE value predicted under the condition of lane keeping maneuvering is lower than the total dataset loss function value. However, when the vehicle is changing the lanes left or right, the loss function value of its trajectory prediction is higher than the overall loss function value. On the NGSIM dataset, the RMSE value of the left and right lane change maneuvers is slightly higher than the RMSE value of the overall predicted trajectory in the first 2 seconds, and the increase in the RMSE value in the next three seconds is higher

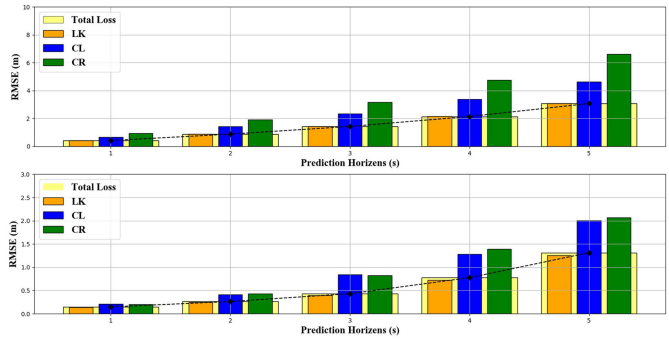


Fig. 9. The average RMSE histogram of EA-Net in the NGSIM and the highD datasets under three maneuvers of lane keeping, left lane change, and right lane change. The yellow column is the overall average RMSE error.

TABLE II
THE STATISTICAL RESULTS OF LANE KEEPING, LEFT LANE CHANGE, AND RIGHT LANE CHANGE 8S TRAJECTORIES IN THE NGSIM AND THE HIGHD DATASETS

Datasets	Maneuver	Train Set	Val Set	Test Set	Total
NGSIM	Lane-Keeping	5659826	829060	1451185	7940071
	Change-Left	204302	24421	39700	268423
	Change- Right	58739	6288	14871	79898
highD	Lane-Keeping	8947120	1238350	2524850	12710320
	Change- Left	243871	33398	71706	348975
	Change- Right	289242	45188	76920	411350

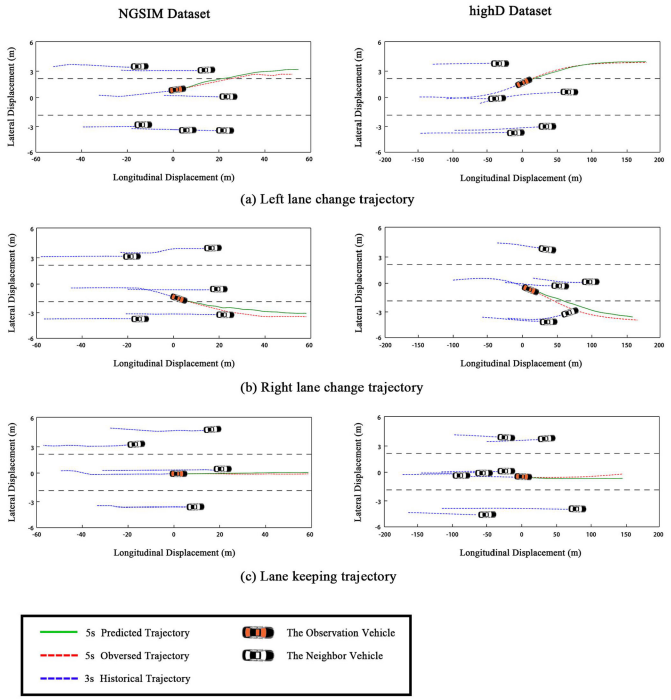


Fig. 10. Vehicle trajectory predictions by the EA-Net model under three maneuvers of lane keeping, left lane change, and right lane change on NGSIM and highD datasets.

than that of the first two seconds. The RMSE loss function values under the right and the left lane change behaviors are about 50% and 25% higher than the overall loss function value, respectively. This is largely related to the lack of lane-changing data in the NGSIM's data distribution. The number of lane-changing trajectories in the dataset is less than the number of lane keeping trajectories, and the number of right-lane-changing trajectories is less than the number of left-lane-changing trajectories. The number of lane-changing trajectories in the highD dataset is also less than the number of lane-keeping trajectories. Among the trajectory loss function values predicted on the highD dataset, the prediction accuracy of the last 3 seconds is lower than that of the first 2 seconds. However, the RMSE values of the left and the right lane change prediction trajectories have a similar increase in 5s, and the loss function values of the left and the right lane change predictions increase every second by about 38% compared to the overall loss function value.

Table II lists the distribution of 8s vehicle trajectories under different maneuvers contained in the both experimental datasets. It can be seen that in the NGSIM dataset, the number of trajectories for the right lane change is smaller than that of the other two maneuvers. In the experiment, under the premise of ensuring that the distribution of each maneuvering trajectory in the training, validation, and test sets is approximately the same, the trajectory distributions in the validation and test sets are slightly adjusted to make the model's test results closer to the real situation. Although the straight trajectory samples accounted for the majority of the highD dataset, the sample sizes of the left and the right lane change trajectories also reached a high value, which enabled the proposed model to obtain sufficient lane change data samples for training in the highD dataset. Thus, a high level model prediction effect is obtained.

C. Prediction Effect Analysis

The trajectories of different maneuvers predicted under the two datasets are visualized and the reliability of the proposed model is analyzed for obtaining environmental feature information by comparing the prediction effects of vehicle trajectories in different traffic environments. Fig. 10 shows the effect diagrams of vehicle trajectory prediction under different maneuvers (straight, left lane, right lane change) under two datasets. The blue dashed lines in the figure represent the 3 seconds historical trajectories of the observed vehicle and its surrounding vehicles, the red dashed line represents the real 5s future trajectory of the observed vehicle, and the green solid line represents the predicted trajectory coordinates of the observed vehicle.

In the two left lane change examples, it can be seen that the model highD dataset has a higher prediction accuracy for the lane change trajectory than the NGSIM dataset. Although the left lane-changing trajectory on the NGSIM dataset has a certain gap with the observed real trajectory, the predicted trajectory is smoother than the original trajectory and tends to be more idealized. In the examples of highD and NGSIM right lane change, the traffic conditions are more complicated, and there are vehicles in front or behind that are closer to the observed vehicle in the target lane. In this case, the future trajectory predicted by the proposed model is closer to the lane line, and the driving distance

is reduced, which means that the vehicle has reduced the driving speed to adapt to the current lane changing behavior. Under the lane keeping behavior, both datasets have high accuracy, and the predicted horizontal and vertical displacements and the implicit vehicle speed are close to the true value.

VI. Conclusion

In this paper, a new neural network model for trajectory prediction is proposed. In terms of trajectory characteristics, the proposed model also considers the relative speed, the acceleration, the heading angle and the relative distance between the vehicles in addition to the x and y coordinates commonly input in the previous studies. In terms of model structure, a new type of environmental feature extraction module is proposed that includes attention mechanism. This module is a parallel structure composed of Graph Attention network and SE-Convolutional social pooling. The attention mechanism and a large number of learnable parameters contained in this module not only enable the proposed model to more efficiently extract the features needed to predict the trajectory, but also effectively amplify the important features.

The proposed model has achieved good results on both NGSIM and highD datasets. Compared with the models including convolutional social pool and graph neural network proposed in the previous studies, the prediction effect of the proposed model on the NGSIM dataset has an accuracy improvement of 27% and 22%, respectively. Compared with the existing model, the prediction effect of the proposed model on the highD dataset is improved by 50%. This shows that considering the spatial structure and graph structure existing in the surrounding environment of the predicted vehicle is indispensable for improving the accuracy of trajectory prediction.

However, since the public datasets used in this paper are all highway datasets and the data collection sites are straight roads, the effect of our model in complex urban traffic scenes remains to be tested. In addition, it is found that effective model input has a positive correlation with the improvement of model prediction effect. In future work, the authors plan to model the urban roads with a certain degree of curvature and complexity, such as intersections, roundabouts, and elevated highways. At the same time, the generalization capabilities of the model in different traffic scenarios will also be considered. Further in-depth research will be conducted to make the future trajectory prediction more in line with the actual application of urban roads under different working conditions.

References

[1] J. Nilsson, J. Silvlin, M. Brannstrom, E. Coelingh, and J. Fredriksson, "If, when, and how to perform lane change maneuvers on highways," *IEEE Intell. Transp. Syst. Mag.*, vol. 8, no. 4, pp. 68–78, Winter 2016.

[2] S. Ulbrich and M. Maurer, "Towards tactical lane change behavior planning for automated vehicles," in *Proc. IEEE 18th Int. Conf. Transp. Syst.*, 2015, pp. 989–995.

[3] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 1468–1476.

[4] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Non-local social pooling for vehicle trajectory prediction," in *Proc. IEEE Intell. Veh. Symp.*, Paris, France, 2019, pp. 975–980.

[5] T. Zhao *et al.*, "Multi-Agent tensor fusion for contextual trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12126–12134.

[6] H. Jeon, J. Choi, and D. Kum, "SCALE-Net: Scalable vehicle trajectory prediction network under random number of interacting vehicles via Edge-enhanced graph convolutional neural network," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2020, pp. 2095–2102.

[7] M. Bahram, C. Hubmann, A. Lawitzky, M. Aeberhard, and D. Wollherr, "A combined model-and learning-based framework for interaction-aware maneuver prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1538–1550, 2016.

[8] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, 2013, pp. 4363–4369.

[9] M. Schreier, V. Willert, and J. Adamy, "Bayesian, Maneuver-Based, long-term trajectory prediction and criticality assessment for driver assistance systems," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst.*, Qingdao, China, 2014, pp. 334–341.

[10] C. Laugier *et al.*, "Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety," *IEEE Intell. Transp. Syst. Mag.*, vol. 3, no. 4, pp. 4–19, Winter 2011.

[11] J. Schlechtriemen, F. Wirthmueller, A. Wedel *et al.*, "When will it change the lane? A probabilistic regression approach for rarely occurring events," in *Proc. IEEE Intell. Veh. Symp.*, Seoul, Korea, 2015, pp. 1373–1379.

[12] Q. Tran and J. Firl, "Online maneuver recognition and multi-modal trajectory prediction for intersection assistance using non-parametric regression," in *Proc. IEEE Intell. Veh. Symp.*, Dearborn, MI, USA, 2014, pp. 918–923.

[13] S. Yoon and D. Kum, "The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2016, pp. 1307–1312.

[14] A. Khosroshahi, E. Ohn-Bar, and M. M. Trivedi, "Surround vehicles trajectory analysis with recurrent neural networks," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst.*, Rio de Janeiro, Brazil, 2016, pp. 2267–2272.

[15] F. Altché and A. de La Fortelle, "An lstm network for highway trajectory prediction," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, Oct 2017, pp. 353–359.

[16] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2018, pp. 1179–1184.

[17] H. Wang, Y. Yu, Y. F. Cai, X. Chen, L. Chen, and Q. C. Liu, "A comparative study of state-of-the-art deep learning algorithms for vehicle detection," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 2, pp. 82–95, Summer 2019.

[18] S. H. Park, B. Kim, C. M. Kang, C. C. Chung., and J. W. Choi, "Sequence-to-Sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture," in *Proc. IEEE Intell. Veh. Symp.*, Changshu, China, 2018, pp. 1672–1678.

[19] H. Jeon, D. Kum, and W. Jeong, "Traffic scene prediction via deep learning: Introduction of multi-channel occupancy grid map as a scene representation," in *Proc. IEEE Intell. Veh. Symp.*, Changshu, 2018, pp. 1496–1501.

[20] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *Proc. IEEE Intell. Veh. Symp.*, Los Angeles, CA, USA, 2017, pp. 204–211.

[21] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang "STGAT: Modeling spatial-temporal interactions for human trajectory prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6272–6281.

[22] Z. Wu *et al.*, "Graph wavenet for deep spatial-temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artificial Intelligence (IJCAI)*, 2019, pp. 1907–1913.

[23] P. Velickovic *et al.*, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Representations*, 2018.

[24] J. Hu, L. Shen, G. Sun, S. Albanie, and E. Wu "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[26] J. Colyar and J. Halkias, "US highway I-80 dataset," Federal Highway Admin, Washington, DC, USA, Tech. Rep. FHW A-HRT-06-137, 2006.

[27] J. Colyar and J. Halkias, "US Highway 101 dataset," Federal Highway Admin, Washington, DC, USA, Tech. Rep. FHW A-HRT-07-030, 2007.

[28] R. Krajewski *et al.*, "The highD dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, Maui, HI, USA, 2018, pp. 2118–2125.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*., 2014.
[30] Y. Yoon *et al.*, "Road-Aware trajectory prediction for autonomous driving on highways," *Sensors*, vol. 20, 2020, Art. no. 4703.
[31] Y. Cai *et al.*, "DLnet with training task conversion stream for precise semantic segmentation in actual traffic scene," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2021.3080261.
[32] Y. Cai *et al.*, "Pedestrian motion trajectory prediction in intelligent driving from far shot first-person perspective video," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TITS.2021.3052908.
[33] Z. Liu *et al.*, "Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2021.3059674.
[34] Q. Qi *et al.*, "Knowledge-driven service offloading decision for vehicular edge computing: A deep reinforcement learning approach." *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4192–4203, May 2019.
[35] Z. Ma *et al.*, "Fine-grained vehicle classification with channel max pooling modified CNNs," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3224–3233, Apr. 2019.
[36] H. Woo *et al.*, "Lane-change detection based on vehicle-trajectory prediction," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 1109–1116, Apr. 2017.

Yingfeng Cai (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the School of Instrument Science and Engineering, Southeast University, Nanjing, China, respectively. In 2013, she joined the Automotive Engineering Research Institute with Jiangsu University, where, she is currently working as a Professor. Her research interests include computer vision, intelligent transportation systems, and intelligent automobiles.

Zihao Wang received the B.S degree from Jiangsu University, Zhenjiang, China. He is working toward the M.S degree with Jiangsu University. His research interests include computer vision, deep learning, and intelligent vehicles.

Hai Wang (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the School of Instrument Science and Engineering, Southeast University, Nanjing, China, respectively. In 2012, he joined the School of Automotive and Traffic Engineering at Jiangsu University, where, he is currently working as a Professor. He has authored or coauthored more than 50 papers in the field of machine vision-based environment sensing for intelligent vehicles. His research interests include computer vision, intelligent transportation systems, and intelligent vehicles.

Long Chen received the Ph.D. degree in vehicle engineering from Jiangsu University, Zhenjiang, China, in 2002. His research interests include intelligent automobiles and vehicle control systems.

Yicheng Li received the Ph.D. degree in vehicle engineering from the Wuhan University of Technology, Wuhan, China, in 2018. He is currently an Assistant Professor with the Automotive Engineering Research Institute, Jiangsu University. His research interests include intelligent vehicle localization, intelligent transportation systems, computer vision, and 3D data processing.

Miguel Angel Sotelo (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Alcalá (UAH), Madrid, Spain, in 2001. He is currently a Full Professor with the Department of Computer Engineering, University of Alcala (UAH). His research interests include autonomous vehicles and prediction of intentions. He was a Project Evaluator, Rapporteur, and Reviewer for the European Commission in the field of ICT for Intelligent Vehicles and Cooperative Systems in FP6 and FP7. He is member of the IEEE ITSS Board of Governors and Executive Committee. He was the Editor-in-Chief of *IEEE Intelligent Transportation Systems Magazine*, the Editor-in-Chief of *ITSS Newsletter*, and an Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. At present, he is the President of IEEE Intelligent Transportation Systems Society.

Zhixiong Li (Senior Member, IEEE) received the Ph.D. degree in transportation engineering from the Wuhan University of Technology, China. he is currently an ARC DECRA Fellow with the School of Mechanical, Materials, Mechatronic and Biomedical Engineering, University of Wollongong, Australia. He is also an Adjunct Professor with the Faculty of Engineering, Ocean University of China. His research interests include Intelligent Vehicles and Control, Loop Closure Detection, and Mechanical System Modeling and Control. He is an Associate Editor of Measurement (Elsevier), Measurement: Sensors (Elsevier), and a Column Editor of *IEEE Intelligent Transportation Systems Magazine.*