Traffic & Transportation

# Vehicle Trajectory Prediction Based on GAT and LSTM Networks in Urban Environments

Xuelong ZHENG[1], Xuemei CHEN[2], Yaohan JIA[3]

Original Scientific Paper
Submitted: 16 Jan 2024
Accepted: 5 Apr 2024

[1] zxlworks@126.com, Beijing Institute of Technology, School of Mechanical Engineering
[2] Corresponding author, chenxue781@bit.edu.cn, Beijing Institute of Technology, Advanced Technology Research Institute; Beijing Institute of Technology, School of Mechanical Engineering
[3] jiayaohan2001@163.com, Beijing Institute of Technology, School of Mechanical Engineering



Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

**ABSTRACT**
Vehicle trajectory prediction plays a critical role before the decision planning of autonomous vehicles in complex and dynamic traffic environments. It helps autonomous vehicles better understand the traffic environments and ensure safe and efficient tasks. In this study, a hierarchical trajectory prediction method is proposed. The graph attention network (GAT) model was selected to estimate the interactions of surrounding vehicles. Considering the behaviour of surrounding agents, the future trajectory of the target vehicle is predicted based on the long short-term memory network (LSTM). The model has been validated in real traffic environments. By comparing the accuracy and real-time performance of target vehicle trajectory prediction, the proposed model is superior to the traditional single trajectory prediction model. The results of this study will provide new modelling ideas and a theoretical basis for the vehicle trajectory prediction in urban traffic environments.

**KEYWORDS**
autonomous vehicle; trajectory prediction; hierarchical; long short-term memory network; graph attention network.

## 1. INTRODUCTION

Autonomous vehicles will occupy an extremely important position in the future intelligent transportation system. Original Entrusted Manufactures, autonomous driving unicorns and leading Internet companies have carried out much applied research on autonomous driving technology. The prerequisite for the safe and efficient driving of intelligent vehicles is good behavioural decision-making abilities. On the other hand, the prerequisite for correct behavioural decision-making is the ability to accurately predict the trajectories of surrounding vehicles [1, 2].

Vehicle trajectory prediction methods are classified into three main categories: physics-based methods [3, 4], classic machine learning-based methods [5, 6] and data-driven methods [7, 8]. Most of the physics-based trajectory prediction methods utilise kinematic models (uniform velocity model, uniform acceleration model and constant pendulum angular velocity and acceleration, etc.) in combination with Kalman filtering to describe the future motion of the target. Barth et al. [9, 10] used the Kalman filter and Monte Carlo methods to predict vehicle trajectories based on vehicle kinematic models, but they cannot be used for trajectory prediction tasks in the long-time domain. Houenou et al. [11] combined the Constant Yaw Angular Velocity and Acceleration (CYRA) model and vehicle behaviour recognition to study the trajectory prediction problem, which compensates for the long-time trajectory prediction problem. Rafael et al. [12] designed a highway lane changing trajectory prediction algorithm based on multi-model interaction considering road shape parameters, which significantly improved the trajectory prediction effect of the vehicle. Some scholars have modelled specific driving behaviours (following, lane-changing, etc.) by building curve-fitting trajectory prediction models. Enke et al. [13] proposed a model for lane-changing trajectories in an ideal state represented by a sine

function but did not consider the influencing factors of the surrounding vehicles and drivers, which resulted in excessive deviation between the predicted trajectories and the actual trajectories. Chovan et al. [14] added a trapezoidal function to the sine function to represent lane-changing trajectories. Kim et al. [15] proposed a trajectory prediction model based on a two-stage trajectory prediction architecture that ensured map-adaptive diversity and accommodated geometric constraints. The autonomous driving team of Daimler Benz [16, 17] used the Kalman filter and Monte Carlo method in combination of vehicle physical model to predict the vehicle trajectories, but this method could not work well during long-term trajectory prediction. Traditional machine learning methods commonly used for trajectory prediction tasks include Gaussian Process (GP), Hidden Markov Models (HMM) etc. HMM is a probabilistic-based framework that can be used to account for the uncertainty of the target's motion pattern [18, 19], but the target is assumed to be an independent individual, and the interactions between agents cannot be taken into account. The GP model [20] is a nonparametric, kernel-function-based probabilistic model that introduces a set of hidden variables obeying a Gaussian distribution to explain the probabilistic prediction problem. The method is capable of generating trajectories with noise points that express the statistical characteristics of the target trajectory distribution. In the literature [21, 22], a Gaussian process model was utilised to predict the probability distribution of a vehicle's future trajectory, and validated on the corresponding dataset. However, constructing a Gaussian process model is complex and the model is non-sparse, requiring input of complete sample or feature information. The research team of the Carnegie Mellon University's School of Robotics [23] proposed a hierarchical trajectory prediction method that combined the Gaussian Mixture Model (GMM) and two-layer Hidden Markov Model (HMM); however, this method is less accurate in trajectory prediction of complex scenes due to the insufficient extraction of target motion features and the limited type and size of training data.

With the rise of deep learning, various types of deep learning networks for temporal prediction have been proposed, and data-driven methods have gradually become a research hotspot in the field of trajectory prediction. Long Short-Term Memory (LSTM) networks are more suitable for trajectory prediction tasks because of their powerful information mining and deep characterisation capabilities. Park et al. [24–26] used the LSTM network as the network basis for trajectory prediction and used the encoder-decoder (Encoder–Decoder) as the sequence generation framework to predict the probability of its future trajectory by inputting the historical trajectory sequence of the target vehicle. Deo et al. [27] comprehensively considered the dynamic interaction problem of the agents using a convolution pooling layer and the future trajectories of vehicles using an improved social LSTM, which improved the accuracy of trajectory prediction. Karatzolou et al. [28] constructed an attention-based sequence-to-sequence (Seq2Seq) trajectory prediction model and verified the effectiveness of the model based on real datasets. However, the above methods can only predict a single trajectory, which makes it difficult to characterise the uncertainty of the agent's motion. Therefore, generative trajectory prediction methods such as Generative Adversarial Networks [29] and Variational Auto-Encoder (VAE) [30] have been successively proposed. Currently, most investigations of trajectory prediction are focused on a single scenario. This modelling approach can only be applied to a certain type of specific driving scene with poor robustness and cannot effectively deal with the dynamic changing actual driving scene. Graph structures have also been used by some scholars for vehicle trajectory prediction tasks. Khandelwal [31] proposed a multimodal behavioural prediction method. Vehicle trajectories and road network information were taken as the model input in the form of a directed graph, and future trajectories were the output based on the contextual features of attention in the graph. Liang et al. [32] proposed a Lane Graph Convolutional Network (LaneGCN) that can effectively capture the complex topology and long-distance dependencies of lanes. This network uses a one-dimensional Convolutional Neural Network (CNN) and GCN to extract the features of the intelligentsia and the map nodes, respectively, and uses spatial attention to capture the interactions between the intelligences and the map, achieving good trajectory prediction results on public datasets. Gao et al. [33] used a vector representation of the scenario with the agent's trajectory and road network structure, and designed a hierarchical network model VectorNet, which avoided lossy rendering and convolutional coding with highly dense computation, and effectively improved the expressiveness and inference speed of the network compared with the CNN-based processing of high-precision maps. Zhao et al. [34] proposed a multimodal trajectory prediction method based on predictive endpoints using VectorNet as the backbone network for scene encoding. The method samples the map and uses multiple MLP layers for target prediction, motion estimation as well as trajectory screening and scoring, and finally obtains multiple possible trajectories for the smart body.

This paper comprehensively considers the influence of the target vehicle trajectories by the lane structure and the dynamic interaction of the surrounding vehicles and divides the vehicle trajectory prediction problem into intention prediction and trajectory generation. A vehicle trajectory prediction model integrating spatio-

temporal features of the scene is proposed based on various deep learning networks such as long-term memory and graph attention, which improves the accuracy of long-term trajectory prediction. In addition, the multiple trajectories generated by the model can not only match the uncertainty of vehicle behaviour intention in various driving scenarios, but also conform to the constraints of map topology and vehicle driving rules with better trajectory acceptability. Based on the real vehicle platform, the detection effect of the vehicle trajectory prediction model is verified, including simulation and real vehicle verification in multiple scenarios. The research contents of this paper are as follows: Section 2 describes methodology, including the introduction of long and short-term memory networks, and graph attention networks. Section 3 describes data extraction and processing. In Section 4, the vehicle trajectory prediction model is given. Section 5 describes the experiment and analyses the results. The conclusion of this study is presented in Section 6.

## 2. MATERIALS AND METHODS

This paper focuses on the interaction effects of the prediction target vehicle and the surrounding vehicles to accomplish the long-term trajectory prediction of the target vehicle. Graph Attention Network (GAT) is characterised by automatic learning and optimisation of connectivity relationships between agents, and can effectively express strong and weak interactions between traffic participants. Long and short-term memory networks (LSTM) can filter useful information and learn long-time and long-distance dependencies to accomplish the prediction of target's long-time position, speed, heading angle, and so on.

### 2.1 Graph attention network

A graph is a data structure formed by a series of interconnected nodes, each node has its own characteristics Fi, and the adjacency matrix A and the degree matrix D are commonly used to describe the structural characteristics of the graph. For a graph with N nodes, the adjacency matrix A is a symmetric matrix of size N. If two nodes *i*, *j* are directly connected, then Aij = Aji = 1, otherwise it is 0. The degree matrix D has the same size as A. Dii represents the number of agents directly connected to agent *i*, and the rest positions are 0.

The GAT network uses the attention mechanism and the adjacency matrix to describe the importance of neighbouring agents for the target agent, it adaptively assigns the weights of neighbouring nodes through the attention mechanism without any type of computationally intensive matrix operations (e.g. inversion) or dependence on the graph structure. This approach can effectively address the inherent drawbacks of spectral graph-based neural networks and make the model suitable for induction and inference problems.

The key to the graph attention network is the graph attention layer, for a graph of N agents, with $h = \{h_1, h_2, h_3, \ldots, h_N\}$ denoting the input features, $h_i \in \mathbb{R}^F$. The output after the graph attention layer is $h' = \{h'_1, h'_2, h'_3, \ldots, h'_N\}$, $h'_i \in \mathbb{R}^{F'}$, where F and F' denote the feature dimensions of the input and output layers, respectively. To calculate the degree of association between two agents as follows [35]:

$$e_{ij} = a(Wh_i, Wh_j) \tag{1}$$

where $a$ denotes linear transformation, $W \in \mathbb{R}^{F' \times F}$, $\mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R}$, it is used to generate more expressive features and then calculates the normalised attention fraction as follows:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(e_{ij}))}{\sum_{k \in N} \exp(\text{LeakyReLU}(e_{ik}))} \tag{2}$$

To make the model more stable, the multi-head attention network [36] is used to weigh the features of each agent. The multiple Header was applied to calculate multiple sets of attention scores separately and to connect their respective results to obtain the final output features as follows:

$$h_i' = \sigma\left(\frac{1}{H}\sum_{h=1}^{H}\sum_{j \in N} \alpha_{ij}^h W^h h_j\right) \tag{3}$$

where $\sigma$ is the activation function and $H$ is the number of heads of the multi-head attention network.

### 2.2 Long and short-term memory network

Predicting the future trajectory of a target based on its historical trajectory is a typical time-series task. Traditional RNN (Recurrent Neural Network) is prone to gradient vanishing problem when dealing with

temporal tasks, and cannot solve the problem of long-term dependence of trajectory sequence. In contrast, the LSTM (Long Short-Term Memory) network can effectively alleviate the gradient disappearance the long-term memory in the temporal task with a unique gating logic.

The LSTM adds a storage cell C to store long-term information in the historical time series based on the RNN and designs three gating units: input gate, output gate and forgetting gate to selectively process cell state information to achieve long-term memory. The gates are a way to let information pass through selectively and are generally implemented by activation functions. Input gates control which new data flows into the storage cell, forgetting gates control information and memory in the storage cell, and output gates control which part of the data is used to calculate the output. The implementation details of the LSTM are as follows [2]:

$$f_t=\sigma\left(W_f\cdot[h_{t-1}, x_t]+b_f\right) \tag{4}$$

where $f_t$ is called the forgetting gate, the $f_t$ is essentially a vector and usually uses sigmoid as the activation function. The output of the sigmoid is a value between the interval [0, 1]. It is defined by:

$$i_t=\sigma(W_i\cdot[h_{t-1}, x_t]+b_i) \tag{5}$$

$$\tilde{C}_t=\tanh(W_C\cdot[h_{t-1}, x_t]+b_C) \tag{6}$$

where $\tilde{C}_t$ denotes the updated value of the cell state, which is determined by the input $x_t$ and the hidden state of the previous moment $h_{t-1}$ obtained via a neural network layer $\tilde{C}_t$. The activation function usually used is tanh. $i_t$ which is called the input gate, and $\tilde{C}_t$ like $x_t$ and $h_{t-1}$ are computed by the activation function as follows:

$$C_t=f_t * C_{t-1}+i_t * \tilde{C}_t \tag{7}$$

where $C_t$ represents the cell state as follows:

$$o_t=\sigma(W_o\cdot[h_{t-1}, x_t]+b_o) \tag{8}$$

$$h_t=o_t * \tanh(C_t) \tag{9}$$

Finally, using the output gate $o_t$ and the above-derived $C_t$ to obtain the state of the hidden layer at the current moment $h_t$. $C_t$ and $h_t$ are passed to the cell at the next moment and the above steps are repeated.

## 3. DATASET AND PROCESSING

### 3.1 Dataset

NGSIM, HighD [37], etc. are captured by high altitude cameras or drones with a fixed bird's eye view, which do not match the real driving scenarios of self-driving vehicles. These data are collected in several fixed scenarios, which limits the scenario diversification of the data. Apollo Scape makes up for the above shortcomings to a certain extent by the data collected by the on-board sensors, but does not include map data, none of the above datasets can meet the requirements of the trajectory prediction model in this paper.

The Argoverse dataset [38], jointly released by Argo AI, the Carnegie Mellon University and others, consists of two parts: 3D tracking and motion prediction. The data was collected mainly in Miami and Pittsburgh, USA, and consisted of sensor data recorded from different seasons, weather and time periods, providing rich real-world driving scenarios. In addition, Argoverse is the first dataset to provide high-precision maps, which contains high-precision map data with geometric and semantic information within 290 km.

### 3.2 Trajectory data processing

In this paper, each dataset is processed into 5-second vehicle scene fragments, containing the trajectories of the predicted object P and the surrounding agents $\{P_i\}$. The vehicle trajectories are observed for a duration $T_{obs}$ = 2 seconds and predicted for a duration $T_{pred}$ = 3 seconds. In this paper, the trajectory sequences are processed by the Savitzky-Golay [38] smoothing filter, which is a filtering method based on least squares fitting.

In terms of trajectory data extraction, the influence of surrounding traffic participants is considered to the highest extent possible. According to the literature [34], the interaction range is set as $R_{agent}$ = 100 m, and the trajectories of the surrounding agents within the interaction range $R_{agent}$ are screened. The trajectories of stationary targets and trajectories with length less than 0.3 $T_{obs}$ are deleted, and the incomplete trajectories are

interpolated and supplemented so that the trajectories of the predicted object $P$ and the surrounding agents $\{P_i\}$ with length $T_{obs}$ are finally obtained. The trajectory data processing flow is shown in *Figure 1*.

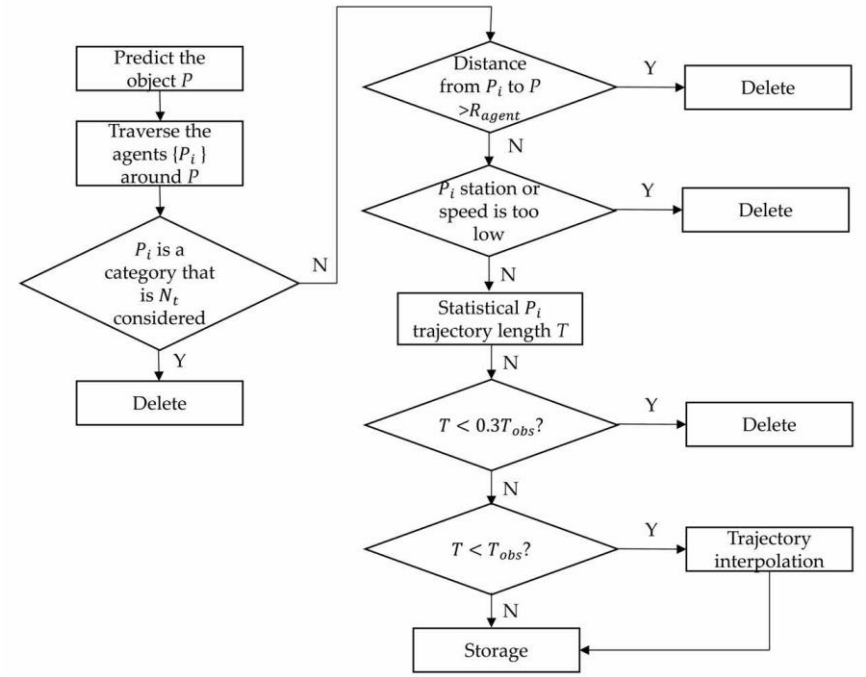



*Figure 1 – Trajectory data processing flow*

### 3.3 Map data processing

In this paper, the map data within a certain range of $R_{lane}$ around the predicted object is extracted as input to the model. In the Argoverse dataset, lane attributes such as lane centreline and lane boundary line are stored. Each training data covers the lane boundary line and lane centreline within $R_{lane}$ = 100 m around the predicted vehicle. To facilitate model training, each lane is represented by a polygon surrounded by 20 points. The lane centreline is used to generate candidate target points for vehicle intent prediction.

After processing the Argoverse dataset, a total of 245,414 vehicle trajectory data are obtained and divided into training set, validation set and test set according to a ratio of 8:1:1. The vehicle trajectory dataset contains 83,721 urban roadway scenes and 161,693 urban intersection scenes.

## 4. MODEL

Autonomous vehicles need to accurately predict the trajectory of surrounding vehicles to help the decision planning system generate an efficient, collision-free local path in real time and ensure a safe and stable operation of the self-driving system. For example, when changing lanes, autonomous vehicles need to predict the future trajectories of the target lane and other vehicles in this lane to ensure the safety of vehicles and drivers. When autonomous vehicles driving on ramps and side roads want to merge into main road, they need to predict the trajectory of the merging traffic to determine the best time. The left turn at unprotected urban intersections requires attention to the movement of oncoming traffic and ensures the safety of the left turn by accurately predicting its future trajectory.

### 4.1 Multi-modal trajectory prediction model

Due to the inherent uncertainty in the behaviour of traffic participants, to ensure safe and efficient driving on the road, autonomous vehicles need to consider multi-modal trajectory prediction of surrounding vehicles. The multi-modality of the trajectory prediction task can be defined as generating multiple acceptable future trajectories that conform to the travel logic and traffic rules based on the uncertainty of the target's behavioural intention. Considering the intersection scenario shown in *Figure 2a*, the target vehicle can possess multiple movement behaviours with different destinations, speeds and curvatures. The autonomous vehicle needs to

fully consider this uncertainty of the target movement, make reasonable predictions of its future behaviours and give probability distributions of the future trajectories.

*Intent prediction based on lane topology.*

With rich lane topology information provided by high-precision maps, the vehicle trajectory prediction problem first extracts a set of candidates predicted trajectory points by searching for future lane sequences of the predicted target. Before searching the lane sequence, it is necessary to construct a directed topology map of lanes based on the high precision map information. Each lane in the high precision map records the IDs of its front and rear left and right lanes. Based on this topological connection information, the road structure of *Figure 2a* is used as an example to create a topological structure map as shown in *Figure 2b*, where the dashed bidirectional arrows indicate that the two lanes are the left and right neighbours. The solid arrows indicate that the two lanes are thefront and rear successors.

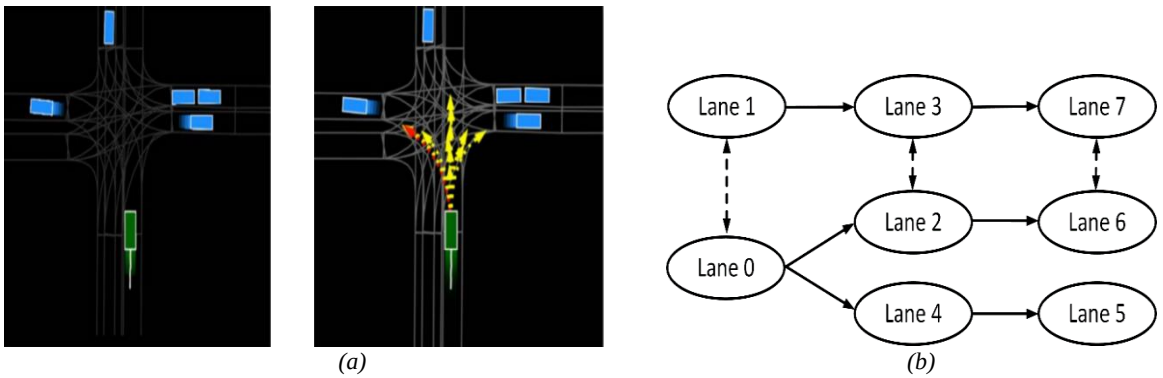



*Figure 2 – (a) Multi-modality for vehicle trajectory prediction (b) Lane topology relationship diagram*

The specific search process for the lane sequence is as follows:

– STEP 1: According to the current position $p_0=(x_0, y_0)$ and direction $\varphi$ of the predicted object $P$, the nearest lane $L_1^1$ of $P$ can be obtained and added to the set $S$ of lane sequences.

– STEP 2: Horizontal (width-first) search. Starting from the lane $L_1^1$, where the object is currently located, the lane connection relationship provided by the high precision map is used to search for the left neighboring lanes $L_1^1$ and right neighbouring lanes $L_1^i (i\neq 1)$ while adding $L_1^i$ to the lane set $S$.

– STEP 3: Vertical (depth-first) search. The depth-first search starts from $L_1^i$ in the set S and searches the successor node $L_1^i$ of each $L_j^i$ and starts the recursion. The final search reaches the set $S$ ($S=\{Seq_i, 1\leq i\leq N_s\}$, $Seq_i=\{L_j^i, j\geq 1\}$) containing $Ns$ Lane sequences, and the recursive exit condition is that the cumulative length of $Seq_i$ exceeds the set search range $R_{lane}$ = 100 m [34]. In *Figure 2b*, the lane sequence obtained after depth-first search with lane 0 as the initial node is $Seq_1=\{L_1^1='0', L_2^1='2', L_3^1='6'\}$ and $Seq_2=\{L_1^2='0', L_2^2='4', L_3^2='5'\}$.

– STEP 4: Iterate through all $Seq_i$ in S, sample the lane centreline equally by $\Delta s$ to get $T_0=\{\tau^n\}_{n=1}^{N_0}=\{(x_n, y_n)\}_{n=1}^{N_0}$, filter $T_0$ by the set maximum. The set of candidate target points $T_{candis}=\{\tau^n\}_{n=1}^{N}$ satisfying $d_{max}\geq dist(\tau^n, p_0)\geq d_{min}$, where $dist(\tau^n, p_0)$ denotes the Euclidean distance between $\tau^n$ and $p_0$, $d_{max}$ and $d_{min}$ are determined by the current velocity $v_{cur}$ of $P$, $d_{max}=(1+\varepsilon)v_{cur}T_{pred}$, $d_{min}=(1-\varepsilon)v_{cur}T_{pred}$, $\varepsilon$ denotes the distance scaling factor, which is used to control the screening range.

$T_{candis}$ represents the possible future location of the predicted object $P$, characterising the uncertainty of the vehicle's motion intention.

*Framework of vehicle trajectory prediction model*

The framework of the vehicle trajectory prediction model constructed in this paper is shown in *Figure 3*. The interaction influence of other agents around the prediction object is also considered. In addition, considering that the vehicle motion is constrained by the road structure, the spatial interaction module adds map information. The whole model should consist of a spatial interaction module, trajectory encoding module, feature fusion module and trajectory output module. The spatial interaction module uses vectors to represent

the spatial interaction of the predicted object with the surrounding agents and the map structure and extracts the spatial interaction features of the predicted object based on the hierarchical network structure. The trajectory encoding module uses the LSTM network to process the historical trajectories (position, velocity, heading angle) of the predicted agents and obtain the temporal features of the historical trajectories. The feature fusion module uses Multi-Head Attention to intersect and fuse spatial interaction features and trajectory timing features. The trajectory output module takes the fused contextual features and the candidate target point set $T_{\text{candis}}$ as input to obtain the multiple possible trajectories and the corresponding probability values. The implementation process of the spatial interaction module and the trajectory output module is described below.

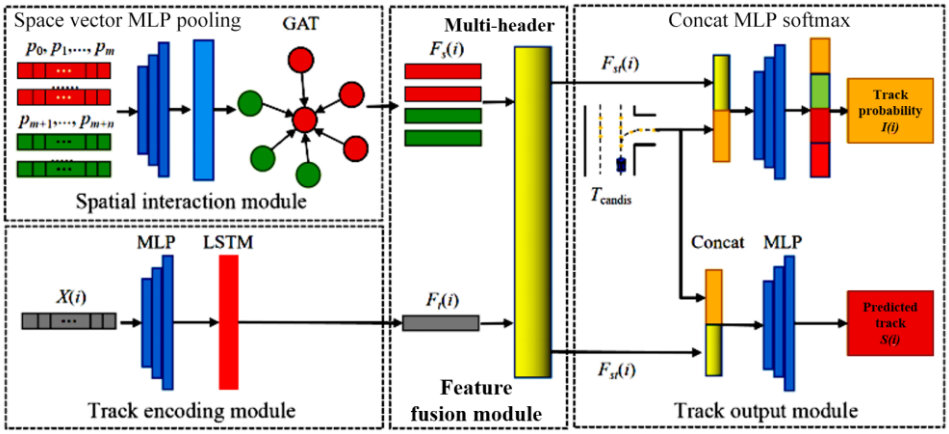



*Figure 3 – Vehicle trajectory prediction model*

1) Spatial interaction module

The framework of Spatial interaction module constructed in this paper is shown in *Figure 4*. Vehicle motion is generally strictly constrained by lane lines. The spatial interaction module uses vectors to represent the trajectories of surrounding agents in addition to lane vectorisation as input. The spatial interaction features $F_{st}(i)$ of the predicted agents are obtained based on the graph attention network (GAT) to model the higher-order interaction between the trajectories and lanes of the agents.

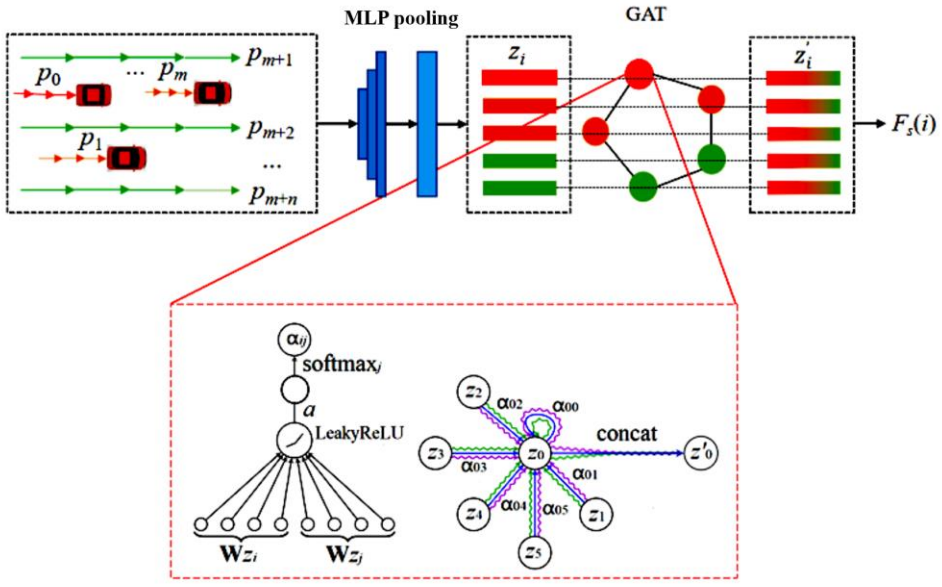



*Figure 4 – Structure of spatial interaction module*

The set of vehicle trajectories can be represented as $\{p_0, p_1, \ldots, p_m\}$. $p_0$ represents the sequence of trajectories of the predicted object. $p_1, \ldots, p_m$ represents the sequence of trajectories of neighbouring vehicles. Each sequence of trajectories is $p_i = \{v_t\}_{t=1}^{T_{obs}-1}$, $i = 0, 1, \ldots, m$. $T_{obs}$ Indicates the length of the historical trajectory, $v_t$ is an 8-dimensional vector:

$$v_t=[x_s, y_s, x_e, y_e, length, width, ts, id] \tag{10}$$

where $x_s, y_s, x_e$ and $y_e$ represent the beginning and end of the spatial location of the vehicle trajectory, length and width represent the length and width of the predicted vehicle, *ts* represents a normalised timestamp, *id* represents the dependency between the vector $v_t$ and the sequence of trajectories $p_i$, same track sequence with same id.

Spatial interaction module is a hierarchical structure consisting of a two-layer MLP and GAT. The two-layer MLP maps the feature vectors to a high-dimensional space, with the dimensionality increasing from 8 to 64, and then employs Max Pooling in the temporal dimension to obtain the following:

$$z_i=\varphi_{agg}\left(\mathrm{MLP}(p_i ; \mathrm{W_{MLP}})\right), i=0, 1, \ldots, \mathrm{m} \tag{11}$$

the higher-order feature vectors $z_i$ corresponding to the trajectory sequence $p_i$:
where $\mathrm{W_{MLP}}$ represents the weight matrix of the MLP layer, $\varphi_{agg}(\cdot)$ represents the maximum pooling operation.

GAT uses the attention mechanism and adjacency matrix to describe the importance of the neighbouring agents to the target vehicle [39]. It improves the expressive power of graph neural networks compared to graph convolutional neural networks (GCN) [40] by adaptively assigning weights to the neighbouring agents through the attention mechanism. Brody et al. [41] propose an improved GAT_v2 network, which overcomes the deficiency of the traditional GAT that can only provide static attention by changing the computation order of the linear layer Linear and the activation function LeakyReLU. In this paper, we use the above improved GAT_v2 to realise the higher-order interactions of the trajectory sequence features $\{z_i\}_{i=0}^{m}$:

$$z_i^{'}=\sigma\left(\frac{1}{H}\sum_{h=1}^{H}\sum_{j=0}^{m}\alpha_{ij}^{h}W^{h}z_j\right) \tag{12}$$

$$\alpha_{ij}=\frac{\exp\left(a\left(\mathrm{LeakyReLU}\left(Wz_i,Wz_j\right)\right)\right)}{\sum_{k=0}^{m}\exp\left(a\left(\mathrm{LeakyReLU}\left(Wz_i,Wz_k\right)\right)\right)} \tag{13}$$

where $z_i^{'}$ represents the trajectory sequence features after the GAT aggregation, with the same dimension of 64. *W* represents the weight matrix of the initialised linear transformation, and *a* is implemented by the MLP layer to compute the similarity between two features. $\sigma$ and LeakyReLU are activation functions and *H* is the number of heads of the network as follows:

$$F_s(i)=\{z_0^{'}, z_1^{'}, \ldots, z_m^{'}\} \tag{14}$$

For the trajectory sequence features $\{z_0, z_1, \ldots, z_m\}$ of the neighboring vehicles, it can be expressed as $\{z_0^{'}, z_1^{'}, \ldots, z_m^{'}\}$ after the higher-order interactions of the GAT, i.e. it is the spatial interaction feature $F_s(i)$.

(1) Track encoding module

Track encoding module is implemented based on LSTM network, which is mainly used to extract the trajectory time-series features of the predicted agents. The multidimensional feature vector $X(i)=\{(x_i^t, y_i^t, v_i^t, \varphi_i^t) \in R^2 | t=1, \ldots, T_{obs}\}$ composed of position, velocity and heading angle is used as input. The temporal encoding property of the LSTM is utilised to extract the temporal features $F_t(i)$ in the historical trajectory of the predicted object. X(i) is mapped into a high-dimensional feature vector $Emb(i)=\{e_i^t\}$, t=1, …, $T_{obs}$, via a two-layer MLP, and then this vector is inputted into an LSTM network, which outputs the trajectory temporal feature $F_t(i)$:

$$\mathrm{Emb}(i)=\mathrm{MLP}(X(i);\mathrm{W_{MLP}})) \tag{15}$$

$$F_t(i)=\mathrm{LSTM}(\mathrm{Emb}(i), h; \mathrm{W_{LSTM}}) \tag{16}$$

where $W_{MLP}$ and $W_{LSTM}$ represent the weight matrices of the MLP layer and the LSTM network, respectively, and h represents the state of the hidden layer of the LSTM network, with a dimension of 64.

2) Feature fusion module

To fuse the spatial interaction feature $F_s(i)$ with the temporal feature $F_t(i)$, previous work often uses various pooling functions or direct splicing to merge the features, which often lacks an effective feature fusion mechanism and may ignore some important interaction information, resulting in a negative impact on the final prediction results. The attention mechanism is a modelling tool that allows the model to focus on key information and fully learn and absorb it. In this paper, the feature fusion module adopts the mainstream multi-attention mechanism to intersect and fuse the features $F_s(i)$ and $F_t(i)$. The trajectory time-series feature $F_t(i)$ is used as the query vector to calculate the interaction strength between the object and other agents. By allocating different degrees of attention, the network pays more attention to the individuals that have a greater influence on the predicted object, and obtains the scene context feature $F_{st}(i)$.

$$F_{st}(i)=\text{MultiAtten}\ (F_t(i),\ F_s(i);\ \text{W}_{\text{MultiAtten}}) \tag{17}$$

$$\text{MultiAtten}\ (Q,\ K,\ V) = \text{Concat}\ (\text{head}_1,\ \text{head}_2,\ldots,\ \text{head}_\text{H})\text{W}^O \tag{18}$$

$$\text{head}_i=\text{Attention}(Q\text{W}_i^Q,\ K\text{W}_i^K,\ V\text{W}_i^V) \tag{19}$$

where $\text{head}_1$, $\text{head}_2$,…, $\text{head}_\text{H}$ represent multiple head structures, $H$ is the number of heads of the network, $W_i^Q$, $W_i^K$, $W_i^V$, $W^O$ are weights matrices, and Concat (·) represents the splicing operation.

The attention weight calculation function uses the scaled-dot product approach [41]:

$$\text{Attention}(Q,\ K,\ V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{20}$$

The multi-headed attention mechanism divides the network into multiple subspaces, which can effectively prevent the network from overfitting. Specifically, the spatial features $F_s(i)$ are used as $K$ and $V$. The trajectory temporal features $F_t(i)$ are used as the query vector $Q$ to adaptively evaluate the degree of association between the predicted object and the surrounding agents.

3) Track output module

The trajectory output module consists of a trajectory classification branch and a trajectory regression branch. By decoding the fused scene context features $F_{st}(i)$ and the set of candidate target points $T_{\text{candis}}=\{\tau^n\}_{n=1}^N$. it outputs multiple possible trajectories and the corresponding probability values of the predicted agents.

The trajectory classification branch consists of a two-layer MLP stacked with softmax layer, and inputs $F_{st}(i)$ and $T_{\text{candis}}$, and the softmax layer outputs the probability distribution $I(i)=\{i^1,\ i^2,\ldots,\ i^N\}$ with "$N \times I$" parameters. $I(i)$ is the output of the module for multiple trajectories, which characterises the uncertainty of the vehicle's motion intention.

$$I(i)=\text{softmax}(\text{MLP}(\ [F_{st}(i),\ T_{\text{candis}}];\ \text{W}_{\text{MLP}})) \tag{21}$$

where [·] represents the splicing operation and $\text{W}_{\text{MLP}}$ represents the network weight of the MLP layer.

The trajectory regression branch utilises a two-layer MLP as a decoder to generate multiple predicted trajectories, with the same inputs $F_{st}(i)$ and $T_{\text{candis}}$ , and the N trajectories are generated as $E(i)=\{\ [E_1^n,\ E_2^n,\ldots,\ E_{T_{\text{pred}}}^n]\}_{n=1}^N$. For each predicted trajectory point $E_t^n$ output the binary mixture Gaussian distribution with five parameters $\mu_{tx}^n$, $\mu_{ty}^n$, $\sigma_{tx}^n$, $\sigma_{ty}^n$, $\rho_t^n$.

$$E(i)=\text{MLP}(\ [F_{st}(i),\ T_{\text{candis}}];\ \text{W}_{\text{MLP}}) \tag{22}$$

where [·] represents the splicing operation and $W_{MLP}$ represents the network weight of the MLP layer.

In order to ensure the multimodality of trajectory prediction and limit the predicted trajectories to a suitable subset, the non-maximum suppression (NMS) mechanism [42] is used to filter the multiple predicted trajectories $E(i)$ according to the probability of the trajectories in descending order, and the module ultimately outputs $K$ trajectories $E^{'}(i)$.

$$E^{'}(i)=\{E^k\}_{k=1}^K=\text{NMS\_select}(E(i);\ Th_{\text{NMS}}) \tag{23}$$

where, $Th_{NMS}$ is the threshold for NMS screening of the predicted trajectories.

The NMS screening process is as follows:
– Step1: Sort the trajectories in *E(i)* from highest to lowest probability.
– Step2: Select the trajectory *E(i)* with the highest probability from the set $E^n$, add it to the set $E^{'}(i)$, and delete it from *E(i)*.
– Step3: Iterate the remaining trajectories in the set *E(i)*, calculate the final displacement error FDE with $Z^n$, if FDE is less than $\mathrm{Th}_{\mathrm{NMS}}$, remove it from *E(i)*.
– Step4: Repeat the above Step1 to Step3 until *E(i)* is empty or the number of trajectories in $E^{'}$(i) reaches K.

### 4.2 Loss function

To ensure the multimodality of vehicle trajectory prediction, the loss functions $L_{\mathrm{cls}}$ and $L_{\mathrm{reg}}$ need to be designed for the classification branch and the regression branch respectively, and the joint loss function of the whole model is constructed by weighting $L_{\mathrm{cls}}$ and $L_{\mathrm{reg}}$.

$$L(\theta)=\alpha L_{\mathrm{cls}}+\beta L_{\mathrm{reg}} \tag{24}$$

where $\alpha$ and $\beta$ denote the weight coefficients of $L_{\mathrm{cls}}$ and $L_{\mathrm{reg}}$.

The trajectory classification loss $L_{\mathrm{cls}}$ is implemented based on the Cross Entroy Loss function, which ensures that the model is trained without modal collapse due to the candidate target point $T_{\mathrm{candis}}$ as a priori information. To find the nearest candidate target point $\hat{\tau}$ from $T_{\mathrm{candis}}=\{\tau^n\}_{n=1}^{N}=\{(x^n,y^n)\}_{n=1}^{N}$ as the label for correct classification, the classification loss $L_{\mathrm{cls}}$ is defined as follows:

$$L_{\mathrm{cls}}=\mathrm{cross\_entroy}(\{\tau^n\}_{n=1}^{N},\hat{\tau}) \tag{25}$$

The trajectory regression loss $L_{\mathrm{reg}}$ of the constructed trajectory prediction model is obtained by taking the negative logarithm of the probability density function of the binary mixed Gaussian distribution.

$$\mathrm{P}\left(x_t^n,y_t^n\mid\mu_t^n,\sigma_t^n,\rho_t^n\right)=\frac{1}{2\pi\sigma_{tx}^n\sigma_{ty}^n\sqrt{1-\left(\rho_t^n\right)^2}}\exp\left[\frac{-Z_t^n}{2\left(1-\left(\rho_t^n\right)^2\right)}\right] \tag{26}$$

$$Z_t^n=\frac{\left(\tilde{x}_t-\mu_{tx}^n\right)^2}{(\sigma_{tx}^n)^2}+\frac{\left(\tilde{y}_t-\mu_{ty}^n\right)^2}{\left(\sigma_{ty}^n\right)^2}-\frac{2\rho_t^n\left(\tilde{x}_t-\mu_{tx}^n\right)\left(\tilde{y}_t-\mu_{ty}^n\right)}{\sigma_{tx}^n\sigma_{ty}^n} \tag{27}$$

$$L_{\mathrm{reg}}=-\frac{1}{NT_{\mathrm{pred}}}\sum_{n=1}^{N}\sum_{t=1}^{T_{\mathrm{pred}}}\log(P(x_t^n,y_t^n\mid\mu_t^n,\sigma_t^n,\rho_t^n)) \tag{28}$$

where $\mu_t^n$, $\sigma_t^n$, $\rho_t^n$ are the binary mixture Gaussian distribution parameters of the *n* trajectory output from the trajectory regression branch at time *t*. *N* is the number of predicted trajectories and $T_{pred}$ represents the trajectory prediction time.

The training environment, the hardware configuration of the vehicle trajectory prediction model, based on the improved vehicle trajectory dataset for model training, and the input data need to be normalised. The initial learning rate of the optimizer is set to 0.0005, the step size of learning rate decay is set to 10, the number of training rounds is 500 and the batch training size is set to 64. Each MLP layer in the model is followed by the *L1* regularisation layer and the activation function ReLU layer, and the final output trajectory number *K* is 3.

## 5. EXPERIMENTS

### 5.1 Model performance validation results

To verify the effectiveness of each module of the vehicle trajectory prediction model, ablation experiments were designed as well. *Table 1* shows the results of trajectory prediction with only spatial interaction module, only trajectory coding module and complete structure, the prediction duration is 3 s and the model finally outputs 3 trajectories ($K$ = 3). The experimental results show that the model with complete structure has the minimum minADE and minFDE, the prediction accuracy is higher than the single module, and the validity of each module of the model is verified. The speeds in the table refer to the time required for each forward propagation of the model on the TESLA V100 with 32 Gigabytes.

*Table 1 – Results of ablation experiments*

| Spatial interaction module | Track encoding module | Feature fusion module | Track output module | minADE/m | minFDE/m | time/ms |
|---|---|---|---|---|---|---|
| | √ | | √ | 1.65 | 2.97 | 6.5 |
| √ | | | √ | 1.19 | 2.21 | 10.7 |
| √ | √ | √ | √ | 1.10 | 1.97 | 11.4 |

## 5.2 Evaluation metrics for trajectory prediction

The common evaluation metrics for trajectory prediction tasks are Average Displacement Error (ADE), Final Displacement Error (FDE) and Recall. The predicted trajectory is denoted as $s = [s_1, s_2, .., s_{T_{\text{pred}}}]$ and the true trajectory is denoted as $s^{\text{gt}} = [s_1^{\text{gt}}, s_2^{\text{gt}}, \ldots, s_{T_{\text{pred}}}^{\text{gt}}]$.

*ADE and minADE*

ADE denotes the average of the Euclidean distance between the predicted and true trajectories throughout the prediction time domain as follows:

$$\text{ADE} = \frac{1}{T_{\text{pred}}} \sum_{t=1}^{T_{\text{pred}}} \left\| s_t - s_t^{gt} \right\| \tag{29}$$

where $\left\| s_t - s_t^{gt} \right\|$ denotes the Euclidean distance between $s_t$ and $s_t^{gt}$.

For multiple prediction trajectories, minADE is defined as follows:

$$\text{minADE} = \min(\text{ADE}_1, \text{ADE}_2, \ldots, \text{ADE}_K) \tag{30}$$

*FDE and minFDE*

FDE denotes the Euclidean distance between the predicted endpoint $s_{T_{\text{pred}}}$ and the true endpoint $s_{T_{\text{pred}}}^{gt}$ as follows:

$$\text{FDE} = \left\| s_{T_{\text{pred}}} - s_{T_{\text{pred}}}^{gt} \right\| \tag{31}$$

For multiple prediction trajectories, minFDE is defined as follows:

$$\text{minFDE} = \min(\text{FDE}_1, \text{FDE}_2, \ldots, \text{FDE}_K) \tag{32}$$

*Recall*

The meanings of *TP*, *FP*, *TN* and *FN* are shown in Table 2. Recall, also known as the check-percentage, indicates how many of the positive samples can be correctly detected, and it is calculated by using the following formula:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{33}$$

*Table 2 – TP, FP, TN and FN meanings*

| | Correct prediction | False prediction |
|---|---|---|
| Genuine label | TP | FN |
| False label | FP | TN |

Unlike image detection, trajectory prediction is a regression task, and the use of the *Recall* metric requires setting a suitable threshold to specify the allowable trajectory prediction error. For *N* trajectory prediction results, the case where the FDE or minFDE of the predicted trajectory is less than the threshold is defined as the correct prediction. *Recall* is defined as follows:

$$\text{Rcall}=\frac{\sum_{n=1}^{N} \text{L}(\text{minFDE}_n, \text{threshold})}{N} \quad (34)$$

where L(·) is an indicator function that takes 1 when $\text{minFDE}_n$ < threshold, and 0 otherwise. *Recall* describes the reliability of the trajectory prediction model.

### 5.3 Model comparison experiment

Unscented Kalman Filter [43] (UKF) estimates the state based on time forward propagation. It assumes a strict vehicle kinematics modelling making it a significant advantage in short-time prediction (typically within 1 second).

VectorNet [33], a vehicle trajectory prediction model based on high-precision maps proposed by Waymo, predicts the future trajectory of the vehicle by considering the spatial interaction features of each scene element. VectorNet is a more mainstream vehicle trajectory prediction model.

STF [44], a Spatial-Temporal Fusion (STF) model including Multi-layer perceptions (MLP) and Graph Attention (GAT), predicts the future trajectory of the vehicle by the spatial and temporal information historical trajectories simultaneously on the 3D graph. The proposed STF outperforms several baseline methods, especially on the long-time-horizon trajectory prediction.

The vehicle trajectory prediction model in this paper is validated against the above two mainstream models. For urban roadway and intersection scenarios, the prediction effects of the three models are evaluated based on the processed Argoverse dataset. The minADE, minFDE and Recall comparisons are shown in *Tables 3 and 4*. For the single trajectory prediction model, minADE is equivalent to ADE and minFDE is equivalent to FDE. recall@2m indicates that the recall threshold is 2 m.

*Table 3 – Vehicle trajectory prediction results for the straight lines*

| Model | Tpred = 2 s | | | Tpred = 3 s | | |
|---|---|---|---|---|---|---|
| | minADE/m | minFDE/m | Recall@2m | minADE/m | minFDE/m | Recall@3m |
| UKF | 1.59 | 2.46 | 0.35 | 2.16 | 3.49 | 0.39 |
| VectorNet | 0.93 | 1.53 | 0.71 | 1.17 | 2.03 | 0.68 |
| Proposed model | 0.57 | 0.92 | 0.93 | 0.92 | 1.67 | 0.82 |

*Table 4 – Vehicle trajectory prediction results for intersection scenarios*

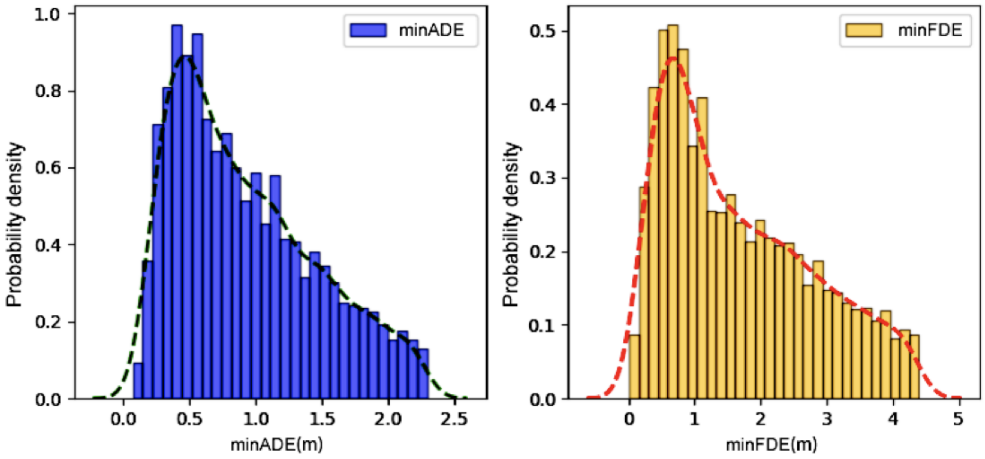
| Model | Tpred = 2 s | | | Tpred = 3 s | | |
|---|---|---|---|---|---|---|
| | minADE/m | minFDE/m | Recall@2m | minADE/m | minFDE/m | Recall@3m |
| UKF | 1.75 | 3.11 | 0.19 | 2.64 | 5.28 | 0.15 |
| VectorNet | 1.12 | 1.97 | 0.48 | 1.95 | 3.90 | 0.35 |
| Proposed model | 0.91 | 1.56 | 0.67 | 1.27 | 2.13 | 0.62 |

The results in *Tables 3 and 4* show that the prediction errors of the UKF model in both scenarios are significantly larger than the other models, and the minFDE at $T_{\text{pred}}$ = 3 s are 3.49 m and 5.28 m. It is difficult to be used for long-time trajectory prediction. The trajectory prediction errors of the VectorNet model have been significantl in y reduced comparison to the UKF, which suggests that data-driven methods are more suitable for long-time trajectory prediction tasks than traditional physics-based methods. The prediction accuracy and recall of the proposed model are better than that of other models. The minFDE of the road section scenario is reduced by 17.7% compared with VectorNet for $T_{\text{pred}}$ = 3 s. The minFDE of the intersection scene is reduced by 28.3% compared with STF. The results indicate that the proposed method of combining maps to predict the target's future trajectory is more accurate and reliable.

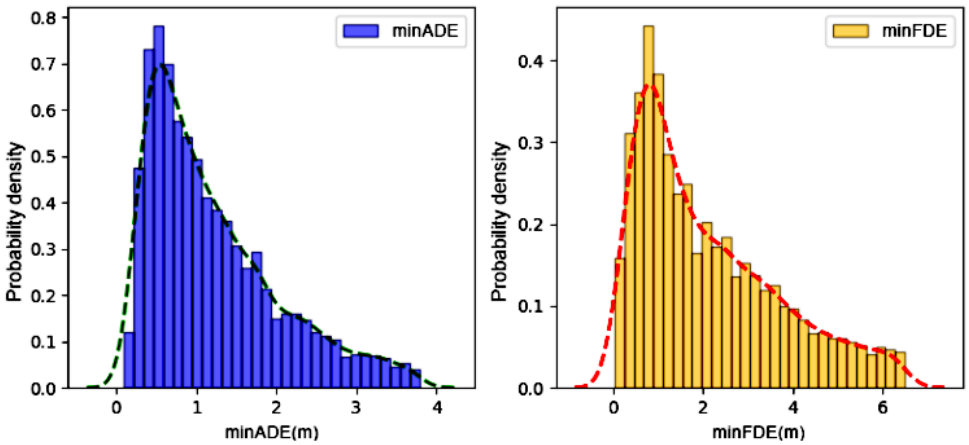
### 5.4 Error distribution of the proposed model

The error distributions of minADE and minFDE of the proposed model in the straight lines and intersection scenarios at Tpred =3 s are shown in Figure 5. It can be found that there are some differences in the error distributions under the two scenarios. The prediction error in the straight lines mainly comes from the longitudinal motion. The mean value of the error is smaller overall. In contrast, the vehicle motion in the intersection scenarios is more difficult to predict, the vehicle crossing the intersection is often accompanied by

speed changes, and the transverse and longitudinal positions also change greatly when turning. The distribution interval of the model minADE and minFDE increases significantly, and the mean value of the error is larger than that of the straight lines.




*(a) Straight Lines*



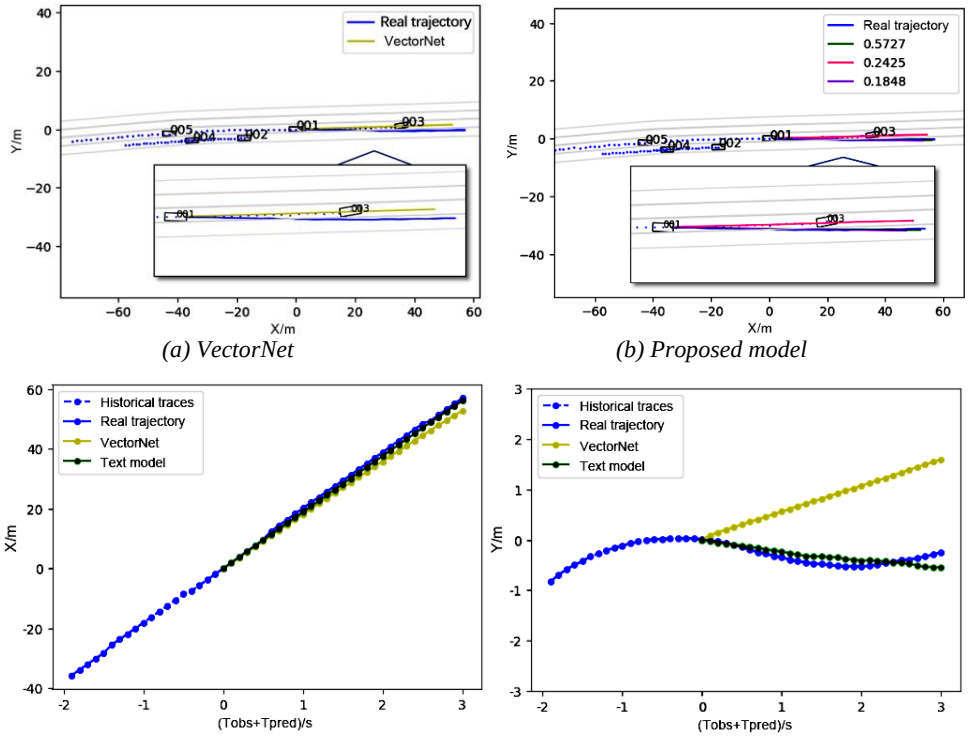

***(b)** Intersection Scene*

*Figure 5 – Distributions of minADE and minFDE*

## 5.5 Trajectory prediction case studies

The trajectory prediction effect of the model is visualised and analysed for typical urban traffic scenes. To verify the effectiveness of the proposed model, VectorNet is selected as a comparison, and the effects of the two models are analysed in urban straight lines and intersection scenes respectively. Each scene shows the predicted trajectories, position changes in *X* and *Y* directions visualised by proposed model and VectorNet, respectively. Multiple predicted trajectories of the proposed model are represented by solid lines of different colours, and each predicted trajectory has a corresponding probability value, while blue dashed lines and blue solid lines indicate the historical trajectories and the future real trajectories of the agent, respectively. When analysing the positional changes in the *X* and *Y* directions, the trajectory with the highest probability of the proposed model is selected for comparison with VectorNet.

Scenario 1 (*Figure 6*) are driving scenarios under straight lines, and the trajectory prediction especially needs to accurately identify the target's lane-changing behaviour. In Scenario 1, car 001 is the predicted vehicle, car 003 in front of this lane at 35 metres is traveling in the same direction with it, car 002 in the right rear has a certain safety distance from this car. There is some chance that the predicted traffic will change lanes to the right. The predicted trajectories of the two models are shown in *Figure 6*. VectorNet misdiagnoses the target's motion intention, and the *Y*-direction position error is significantly larger than that of the proposed model. In contrast, the proposed model accurately predicts the target vehicle's intention, and outputs a trajectory with the highest probability of changing lanes to the right (green solid line), and this trajectory line is the closest to the real trajectory, which indicates that the proposed model not only accurately recognises the intention of the vehicle, but also predicts a more accurate trajectory.

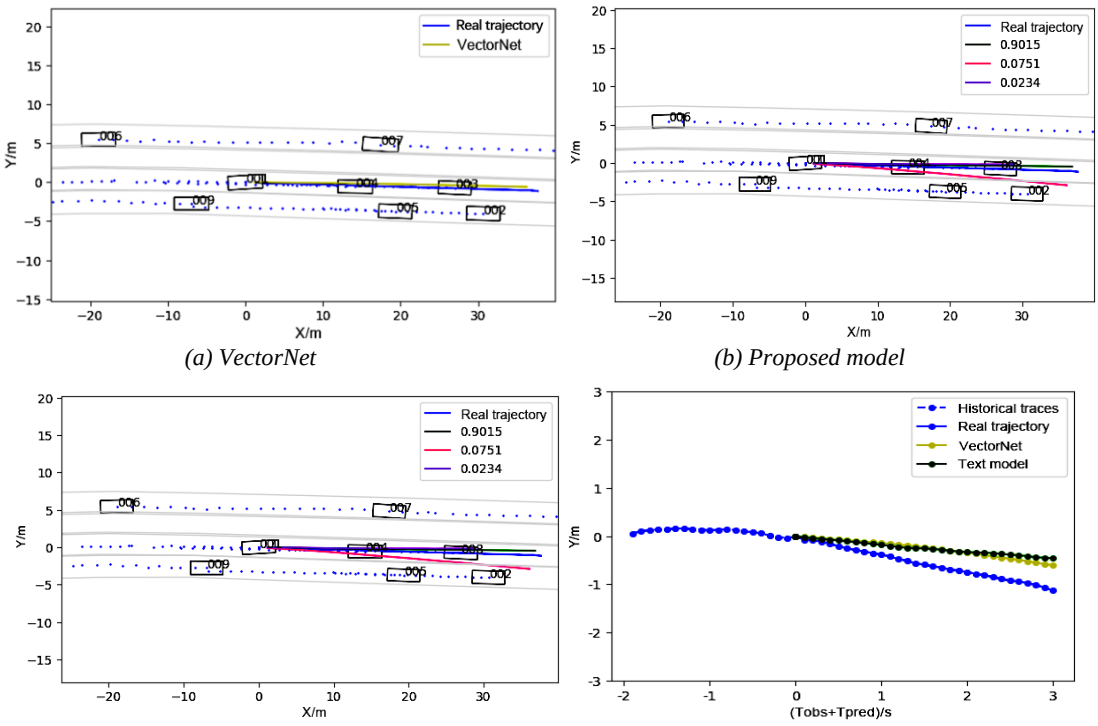



*(a) VectorNet* *(b) Proposed model*

*(c) X-direction prediction error* *(d) Y-direction prediction error*

*Figure 6 – Scenario 1 trajectory prediction results*

Scenario 2 (*Figure 7*) has higher traffic density and fewer lane changing opportunities. The proposed model outputs two straight trajectories and one trajectory of changing lanes to the right. The longitudinal difference of the straight trajectories reflects the two speed intentions of the target, in which a straight trajectory with probability 0.90 is closer to the real trajectory. This indicates that the proposed model can also achieve better prediction of the target's speed, and the corresponding probability of the trajectory of changing lanes to the right is only 0.07, which is less referential for the practical application. *Figure 7(c),(d)* shows that both models have better results in this simple following scenario, and the trajectory prediction errors are all smaller.

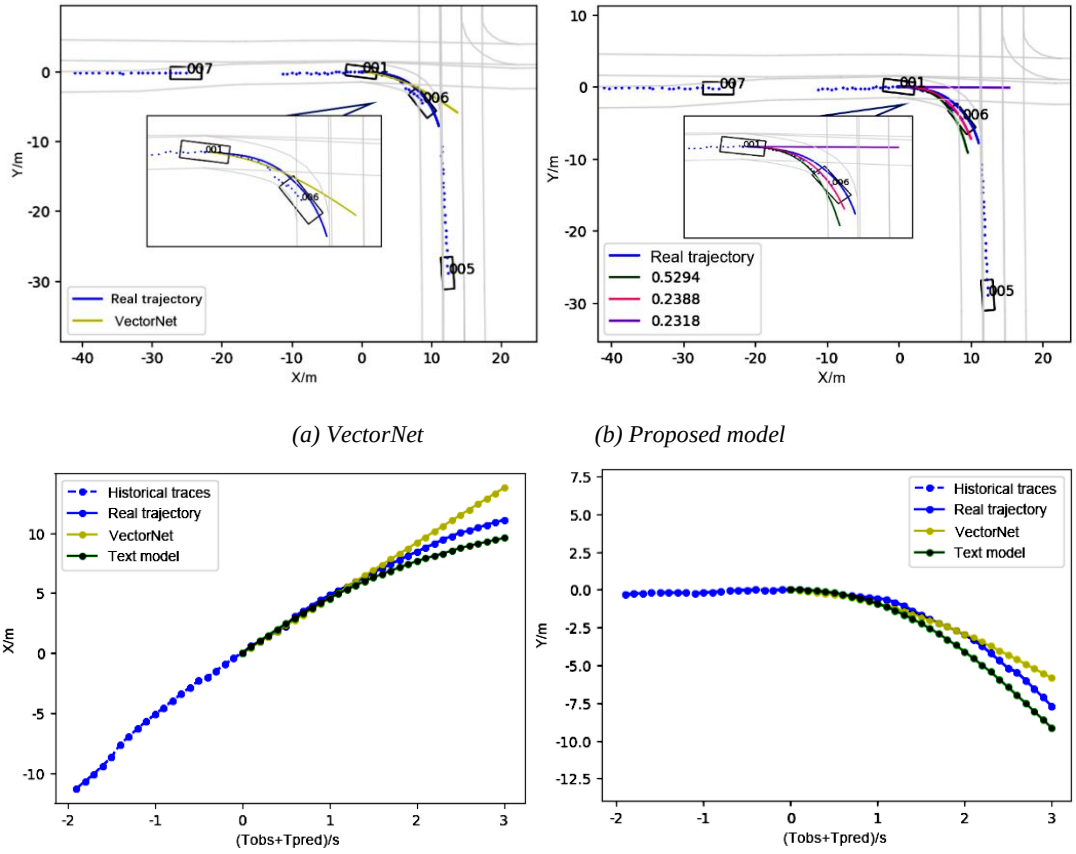



*(a) VectorNet* *(b) Proposed model*

*(c) X-direction prediction error* *(d) Y-direction prediction error*

*Figure 7 – Scenario 2 trajectory prediction results*

The predicted vehicle in Scenario 3 is about to enter the intersection, and from the map structure, the vehicle can either go straight or turn right. VectorNet is unable to capture the multimodality of the scenario, and the predicted trajectories cross multiple lanes and tend to be the average of the two behavioural intents of going straight and turning left, which is obviously against the rules of vehicular travel. The proposed model combined with the map structure can effectively capture multiple possible intentions of the predicted vehicle, outputting one straight ahead trajectory and two right turn trajectories. The probability of one of the right-turn trajectories is 0.52, which indicates that the target is more likely to make a right turn, and the target's real behaviour is to make a right turn. The results validate the accuracy of the proposed model for the prediction of intentions. *Figure 8c,d* shows that the errors of VectorNet in the *X* and *Y* directions when predicting 3 seconds are 2.62 m and 2.05 m, and the proposed model are 1.60 m and 1.58 m, respectively.




*(a) VectorNet* *(b) Proposed model*

*(c) X-direction prediction error* *(d) Y-direction prediction error*

*Figure 8 – Scenario 3 trajectory prediction results*

## 6. CONCLUSION

A vehicle trajectory prediction model integrating spatio-temporal features in complex urban scenes and a hierarchical trajectory prediction model were constructed based on lane topology. The proposed model describes the uncertainty in the intention of vehicle motion by searching for a sequence of lanes and extracting the lane centroids to provide a priori information for generating trajectories. Then, a multi-modal trajectory prediction model predicts future vehicle trajectory by considering the behavioural intentions, the lane structure and the dynamic interaction of surrounding agents. The joint loss function is designed considering both classification loss and regression loss. The proposed model was trained and validated based on the Argoverse dataset. It reduces the minFDE by 17.7% and 45.4% compared to VectorNet in straight lines and intersecting scenarios. The results indicate that the model has both higher accuracy and better robustness.

Future research should consider the potential effects of self-vehicle decision planning more carefully. The data flow between trajectory prediction and decision planning is unidirectional. Both of them should form a closed loop. Decision planning can help the self-vehicle to achieve more accurate trajectory prediction of the other vehicle, and trajectory prediction combined with gaming is a direction worth considering. Furthermore, practical applications have higher requirements for the efficiency and real-time performance of model

inference, which can be optimised in the following aspects: choosing an efficient and lightweight network structure based on guaranteeing accuracy; utilising model pruning and the TensorRT inference acceleration framework; deploying the model to embedded devices to improve the speed of the network forward inference; further evaluating and filtering the predicted objects, and utilising asynchronous operations to achieve parallel prediction tasks.

## ACKNOWLEDGEMENTS

This work was supported in part by the Key R&D Program of Shandong Province (No2023CXPT032), Taishan Industrial Experts Program, Shandong Provincial Natural Science Foundation (No.ZR2023MF0766）

## REFERENCES

[1] Jiang Y, et al. Vehicle trajectory prediction considering driver uncertainty and vehicle dynamics based on dynamic Bayesian network. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2023;53:689–703. DOI: 10.1109/TSMC.2022.3186639.

[2] Wang K, et al. LSTM-based prediction method of surrounding vehicle trajectory. *2022 International Conference on Artificial Intelligence in Everything (AIE), Lefkosa, Cyprus*. 2022. p. 100-105. DOI: 10.1109/AIE57029.2022.00026.

[3] Huang B, Li K, Wang J. Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using interactive multiple models. *IEEE Transactions on Industrial Electronics*. 2017;65(7):5999–6008. DOI: 10.1109/TIE.2017.2782236.

[4] Wang Y, et al. Trajectory planning and safety assessment of autonomous vehicles based on motion prediction and model predictive control. *IEEE Transactions on Vehicular Technology*. 2019;68(9):8546–8556. DOI: 10.1109/TVT.2019.2930684.

[5] Wang Y, Wang C, Zhao W, Xu C. Decision-making and planning method for autonomous vehicles based on motivation and risk assessment. *IEEE Transactions on Vehicular Technology*. 2021;70(1):107–120. DOI: 10.1109/TVT.2021.3049794.

[6] Zhang S, Zhi Y, He R, Li J. Research on traffic vehicle behavior prediction method based on game theory and HMM. *IEEE Access*. 2020;8:30210–30222. DOI: 10.1109/ACCESS.2020.2971705.

[7] Mercat J, et al. Multi-head attention for multi-modal joint vehicle motion forecasting. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. *IEEE.* 2020. p. 9638–9644.

[8] Hasan F, Huang H. Mals-net: A multi-head attention-based lstm sequence-to-sequence network for socio-temporal interaction modelling and trajectory prediction. *Sensors*. 2023;23(1):530. DOI: 10.3390/s23010530.

[9] Barth A, Franke U. Where will the oncoming vehicle be the next second? *2008 IEEE Intelligent Vehicles Symposium (IV).* New York: IEEE. 2008:1068–1073.

[10] Carvalho A, et al. Stochastic predictive control of autonomous vehicles in uncertain environments. *12th International Symposium on Advanced vehicle control*. 2014:712–719.

[11] Houenou A, et al. vehicle trajectory prediction based on motion model and maneuver recognition. *2013 IEEE International Conference on Intelligent Robots and Systems.* New York: IEEE.2013;4363–4369.

[12] Rafael TM, Miguel ZI. IMM-based lane-change prediction in highways with low-cost GPS/INS. *2009 IEEE Transactions on Intelligent Transportation Systems*, 2009;10(1):180–185. DOI: 10.1109/TITS.2008.2011691.

[13] Enke K. Possibilities for improving safety within the driver vehicle environment control loop. *7th International Conference on Experimental Safety Vehicles Proceeding*. 1979;789–802.

[14] Chovan JD. *Examination of lane change crashes and potential IVHS countermeasures*. America: National Highway Traffic Safety Administration, 1994.

[15] Kim S, Jeon H, Choi JW, Kum D. Diverse multiple trajectory prediction using a two-stage prediction network trained with lane loss. *IEEE Robotics and Automation Letters*. 2023;8(4):2038–2045.

[16] Barth A, Franke U. Where will the oncoming vehicle be the next second? *2008 IEEE Intelligent Vehicles Symposium (IV). New York: IEEE*. 2008. p. 1068–1073.

[17] Carvalho A, et al. Stochastic predictive control of autonomous vehicles in uncertain environments. *12th International Symposium on Advanced Vehicle Control*. 2014. p. 712–719.

[18] Tay C, Mekhnacha K, Laugier C. Probabilistic vehicle motion modeling and risk estimation. *Handbook of Intelligent Vehicles*. 2012;1479–1516. DOI: 10.1007/978-0-85729-085-4_57.

[19] Streubel T, Hoffmann KH. Prediction of driver intended path at intersections. *2014 IEEE Intelligent Vehicles Symposium Proceedings. IEEE*. 2014. p. 134–139.

[20] Guo Y, et al. Modeling multi-vehicle interaction scenarios using gaussian random field. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE.* 2019. p. 3974–3980.

[21] Goli SA, Far BH, Fapojuwo A. Vehicle trajectory prediction with gaussian process regression in connected vehicle environment. *2018 IEEE Intelligent Vehicles Symposium. Piscataway: IEEE Press*. 2018. p. 550–555.

[22] Tran Q, Firl J. Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression. *2014 IEEE Intelligent Vehicles Symposium Proceedings. IEEE*. 2014. p. 918–923.

[23] Li J, et al. Generic probabilistic interactive situation recognition and prediction: From virtual to real. *21st International Conference on Intelligent Transportation Systems (ITSC). IEEE*, 2018. p. 3218–3224.

[24] Altchéa F, Fortelle AL. An LSTM network for highway trajectory prediction. *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). Yokohama: IEEE Press*. 2017. p. 123–128.

[25] Kim BD, et a1. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). Yokohama: IEEE Press.* 2017. p. 399–404.

[26] Park SH, et al. Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture. *2018 IEEE Intelligent Vehicles Symposium(IV). New York: IEEE Press*. 2018. p. 1672–1678.

[27] Deo N, Trivedi MM. Convolutional social pooling for vehicle trajectory prediction. *2018 IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE Press*. 2018. p. 1468–1476.

[28] Karatzolou A, Jablonski A, Beigl M. A seq2seq learning approach for modeling semantic trajectories and predicting the next location. *The 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York: ACM Press.* 2018. p. 528–531.

[29] Goodfellow I, et al. Generative adversarial nets. *Conference and Workshop on Neural Information Processing Systems. New York: Curran Associates Press.* 2014. p. 2672–2680.

[30] Kipf T, Welling M. Variational graph auto-encoders. *Arxiv Preprint*, 2016. DOI: 10.48550/arXiv.1611.07308.

[31] Khandelwal S, et al. What-if motion prediction for autonomous driving. *Arxiv Preprint*, 2020. DOI: 10.48550/arXiv.2008.10587.

[32] Liang M, et al. Learning lane graph representations for motion forecasting. *European Conference on Computer Vision. Germany: Springer*. 2020. p. 541–556.

[33] Gao J, et al. VectorNet: Encoding HD maps and agent dynamics from vectorized representation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2020. p. 11525–11533.

[34] Zhao H, et al. TNT: target-driven trajectory prediction. *Conference on Robot Learning*. 2020.

[35] Velikovi P, et al. Graph attention networks. *International Conference on Learning Representations, Vancouver, Canada*. 2018.

[36] Vaswani, A et al. Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*. December 2017. p. 6000–6010.

[37] Krajewski R, et al. The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems. *21st International Conference on Intelligent Transportation Systems (ITSC). IEEE*. 2018. p. 2118-2125.

[38] Chang MF et al. Argoverse: 3d tracking and forecasting with rich maps. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019. p. 8748-8757.

[39] Winkler M . What is a Savitzky-Golay Filter?. *IEEE Signal Processing Magazine*. 2011;4:111–117.

[40] Monti F, Otness K, Bronstein MM. MOTIFNET: a motif-based graph convolutional network for directed graphs. *2018 IEEE Data Science Workshop (DSW), Lausanne, Switzerland*. 2018. p. 225–228.

[41] Brody S, Alon U, Yahav E. How attentive are graph attention networks. *International Conference on Learning Representations, Vienna, Austria.* 2021.

[42] Jiang B, et al. Acquisition of localization confidence for accurate object detection. *Computer Vision – ECCV*. 2018;816–832. DOI:10.48550/arXiv.1807.11590.

[43] Lefkopoulos V, Menner M, Domahidi A, Zeilinger MN. Interaction-aware motion prediction for autonomous driving: A multiple model Kalman filtering scheme. *IEEE Robotics and Automation Letters*. 2021;6(1):80–87. DOI: 10.1109/LRA.2020.3032079.

[44] Penngian H, et al. STF: Spatial temporal fusion for trajectory prediction. *Computer Vision and Pattern Recognition*. 2023. DOI: 10.1109/M2VIP58386.2023.10413434.

郑雪龙，陈雪梅，贾尧涵

## 基于 GAT 与 LSTM 网络的城市环境下车辆轨迹预测

**摘要**

在复杂多变的交通环境中，车辆轨迹预测对自动驾驶车辆的决策规划起着至关重要的作用。它有助于自动驾驶车辆更好地理解交通环境，确保安全高效地完成任务。本研究提出了一种分层轨迹预测方法。选择图注意力网络（GAT）模型来估计周围车辆的相互作用。考虑到周围代理的行为，基于长短期记忆网络（LSTM）预测目标车辆的未来轨迹。该模型已在真实交通环境中得到验证。通过比较目标车辆轨迹预测的准确性和实时性，所提出的模型优于传统的单一轨迹预测模型。该研究成果将为城市交通环境下的车辆轨迹预测提供新的建模思路和理论依据。

**关键词：**

自动驾驶车辆；轨迹预测；分层；长短期记忆网络；图注意网络