TraGCAN: Trajectory Prediction of Heterogeneous Traffic Agents in IoV Systems

Jie Li, *Member, IEEE*, Han Shi, Yue Guo, Guangjie Han, *Fellow, IEEE*, Ruiyun Yu, *Member, IEEE*, and Xingwei Wang

***Abstract*—As a core component of the Internet of Vehicles, reasoning about the trajectory of pedestrians or vehicles in complex road conditions plays a critical role in autonomous driving and socially aware robotic navigation. Most existing methods do not adequately consider the effects of heterogeneous traffic agents. Toward this end, we propose the traffic trajectory prediction algorithm based on the convolutional attention network (TraGCAN) to predict the trajectories of heterogeneous traffic agents in dense traffic. The algorithm of the proposed method examines the behavior of different traffic agents in terms of both time and space dimensions to identify their movement patterns and interactions. We construct the spatial relationship of traffic agents as a graph structure and introduce a graph convolutional network to extract spatial interactions. In addition, we design a spatial attention mechanism to adaptively calculate weights for all spatial interactions to capture different influences from neighboring agents. To improve the accuracy of trajectory prediction, the algorithm considers the influence of the heterogeneous characteristics of traffic agents on their motion behaviors. We evaluated the performance of the proposed TraGCAN on heterogeneous traffic data sets, and the results demonstrate that the error of TraGCAN is reduced by 15% compared to existing methods.**

***Index Terms*—Autonomous driving, graph convolutional network (GCN), Internet of Vehicles (IoV), trajectory prediction.**

I. Introduction

As AN important application of Internet of Vehicles (IoV), the task of autonomous driving is inseparable from accurate location information of road traffic participants [1]. Predicting the future behavior of traffic participants is an important part of autonomous driving. Autonomous driving technology considering the intelligence of a single vehicle agent attempts to reduce the number of traffic accidents and traffic jams [2], [3], [4], [5]. According to statistics, there were approximately 2.25 million traffic accidents recorded in the United States from February 2016 to March 2019 [6]. Most accidents involve more complex causes, e.g., the unpredictability of the behavior of other road users or driving too quickly. Therefore, vehicles are able to intelligently make judgments based on its surroundings, which is in line with the basic characteristics of the Internet of Things (IoT). In other words, it is critical to accurately predict the behavior of other traffic agents, e.g., cars, bicycles, and pedestrians.

Learning socially aware motion representations is at the core of recent advances in multiagent problems [7]. Behavioral prediction refers to the prediction of the trajectory of an object over a short period of time based on the past trajectory of the object, e.g., pedestrian trajectory and vehicle trajectory prediction tasks. Trajectory prediction has been studied extensively. However, in the case of autonomous driving, the scale of the problem is large, the time and space of the traffic agents depend on dynamic changes, the interaction between the traffic agents and road environment is very complex, and the trajectories of agents are difficult to predict. The complexity of trajectory prediction is reflected in two aspects: 1) different types of traffic agents have different movement characteristics and 2) the random movement of other road users and pedestrians influence the agent movement of traffic agents.

Benefiting from the high computing, caching and data storage capabilities provided by advanced vehicle hardware and vehicle infrastructure [8], machine learning, edge computing, and 5G technologies have been applied to vehicles [9], [10], [11], which has attracted the attention of many scholars [12], [13] and laid a foundation for the deployment of vehicle trajectory prediction methods. Information collection devices, such as cameras, various sensors in the IoT can help autonomous vehicles track the tracks of other road agents to accurately perceive the surrounding environment, thereby predicting their future trajectories. Currently, deep learning techniques are employed in this field to simulate interactions between different traffic agents. For example, social long short-term memory (social LSTM) is a classic trajectory prediction method that takes pedestrians as a unit using a recursive model. In addition, the social LSTM method integrates the hidden state of other pedestrians by adding a pooling layer. We consider that these models are limited to recursive architectures and ignore the value of road space information relative to

Manuscript received 22 June 2022; revised 29 September 2022; accepted 8 December 2022. Date of publication 13 December 2022; date of current version 7 April 2023. This work was supported in part by the National Key Research and Development Projects under Grant 2019YFB1802800; in part by the Liaoning Province Science and Technology Fund Project under Grant 2020MS086; in part by the Shenyang Science and Technology Plan Project under Grant 20206424; in part by the Fundamental Research Funds for the Central Universities under Grant N2116014 and Grant N180101028; and in part by the National Natural Science Foundation of China under Grant 62072094 and Grant 61872073. *(Corresponding author: Guangjie Han.)*

Jie Li, Han Shi, and Xingwei Wang are with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: lijie@mail.neu.edu.cn; 2001800@stu.neu.edu.cn; wangxw@mail.neu.edu.cn).

Yue Guo and Ruiyun Yu are with the Software College, Northeastern University, Shenyang 110819, China (e-mail: guoyue7102@163.com; yury@mail.neu.edu.cn).

Guangjie Han is with the Department of Information and Communication Systems, Hohai University, Changzhou 213022, China (e-mail: hanguangjie@gmail.com).

Digital Object Identifier 10.1109/JIOT.2022.3228818

trajectory mining. A convolutional neural network is employed in CS-LSTM [14] to capture the relative position of local vehicles to make up for a disadvantage of the long short-term memory (LSTM) technique, i.e., the relative position cannot be perceived, which overcomes these limitations to a certain extent.

There are two problems with this approach. First, trajectory prediction in IoT involves a single agent type; however, multiple agent types exist in real-world scenarios, e.g., vehicles, bicycles, and pedestrians. The movement rules of different agents have their own characteristics and differences. In fact, considering the road heterogeneous agent moving state and predicting the trajectory of heterogeneous agent have more practical meaning. Second, when mining the spatial interaction relationship between traffic agents, the features of surrounding road users are not fully utilized, and feature extraction techniques are not sufficiently intuitive; thus, it is difficult to explain the physical meaning of feature states. Therefore, we have designed the TraGCAN algorithm to overcome the above two limitations.

Contribution: We propose a framework for predicting the trajectories of heterogeneous traffic agents based on spatio-temporal data (called TraGCAN), which models the influences on agents from other traffic agents to predict agent trajectories in both the temporal and spatial dimensions. First, to highlight the differences between agents, the model accounts for both agent class features and individual features (size, speed, etc.) Then, in order to describe the spatial interactions between agents, we apply a graph convolutional network (GCN) to the spatial relational connectivity graph to extract complex spatial interactions. Also, in order to ensure the effective transfer of dynamic information in continuous time, we design a dynamic attention mechanism that adaptively assigns weights to different spatial interactions and dynamically updates the overall motion features of surrounding agents. Finally, we analyze the algorithm performance leveraging several publicly available autonomous driving trajectory data sets, and design and complete comparison and ablation experiments. The experimental results show that our model can predict reasonable trajectories in different scenarios.

The remainder of this article is organized as follows. In Section II, we introduce previous work related to trajectory prediction algorithms. Section III describes the preparatory work required for modeling. Section IV describes the design of the proposed heterogeneous traffic trajectory prediction algorithm in detail. In Sections V and VI, the validity of the proposed algorithm is verified in comparative and ablation experiments, respectively. Finally, this article is concluded in Section VII, including suggestions for potential future work.

II. Related Work

Trajectory prediction in IoT scenarios is a typical time series. Several conventional methods were used to analyze time series problems, e.g., the Bayesian network [15], [16], hidden Markov model [17], [18], Gaussian process [19], [20], and random forest classification [21] techniques.

Due to the complex and changeable behaviors of traffic agents, traditional algorithms cannot solve the trajectory prediction problem well. After the recurrent neural network (RNN) was proposed, some studies improved the accuracy of long-term trajectory prediction results using improved RNN algorithms. LSTM-based methods are becoming used increasingly in time series analysis. LSTM can effectively capture the complex motion characteristics of objects. Cheng et al. [22] found that the error of the experimental results of the LSTM-based model was much smaller than that of the CNN-based model, especially, when predicting the long-term trajectory of the aerial vehicles, which further demonstrated the effectiveness of the LSTM-based model in the trajectory prediction task. Altché and de La Fortelle [23] proposed an LSTM-based neural network architecture to predict vehicle trajectories on a highway, which is very critical for safe automatic overtaking or lane changing, and the prediction accuracy of this method was significantly higher than that of previous advanced technologies. The focus of Khosroshahi et al. [24] is on the problem of trajectory classification in crossings. By performing behavioral analysis and modeling with a multilayer LSTM, the model is equipped with abstract representation capabilities. The experimental results confirm that fine-grained classification labels enable trajectory prediction results to be closer to the ground truth. S-LSTM [25] based on an improved LSTM model learns pedestrian movements and predicts their future trajectories. The model introduces the concept of social pooling for the first time, which solves the problem that mutually independent LSTMs cannot capture the complex spatial interactions of pedestrians.

Each agent affects the trajectories of other neighboring agents in the context of IoT. Establishing an effective method to consider the temporal relationships and spatial interactions of agent trajectory sequences simultaneously is a significant problem that must be solved. Lin et al. [26] proved that the spatio-temporal features of traffic flow can be predicted. For delay-sensitive urban traffic, the trajectory can be accurately planned before the traffic starts, which provided a theoretical basis for spatio-temporal trajectory prediction. Xu et al. [27] designed a crowd interaction deep neural network, employed LSTM to extract the features of all pedestrian motion information, and employed the inner product between features to measure the degree of mutual influence between pedestrians. In addition, Chang et al. [28] used map information, e.g., lane direction, drivable area, and elevation height, to track a target moving object, mine the spatial information on a vector map, and improve object tracking accuracy. Xue et al. [29] introduced the factor of the pedestrian's scene to the existing S-LSTM model, which considers the influence between adjacent pedestrians, in an attempt to completely simulate the interaction between people in a city. Deo and Trivedi [14] proposed an LSTM encoder–decoder structure that employs convolutional social pooling to improve the social pooling layer in order to speculate on the interdependence of adjacent vehicles. Xie et al. [30] merged two CNN and LSTM networks, and they optimized the model using a grid search algorithm to achieve double-precision prediction in terms of both time and space dimensions.

In recent years, graph convolution technology has attracted the attention of many scholars. For example, Social-STGCNN (S-STGCNN) [31] adopted a different research concept from the previous trajectory prediction method. This model used a GCN to replace the interactive collection mechanism of the original model and uses a TCN to replace the recursive architecture of the existing model. Wu et al. [32] combined the GCN and GRU to establish a spatiotemporal prediction model of joint trajectories, and they topologically associated the joints to better describe the influence of the spatial relationship between the manipulator joints on the joint trajectory.

Note that the interaction between agents will vary depending on different factors, e.g., the distance between agents. The advantages of attention models in deep learning have become increasingly significant. For example, the attention mechanism can obtain the required information, assign a higher weight to important features, and reduce the impact of unimportant features on the results. Vemula et al. [33] proposed a social attention model to capture the relative influence of each pedestrian in a crowd on the future behavior of another pedestrian, thereby improving trajectory prediction accuracy. Social ways [34] further improved the S-LSTM and Social GAN (S-GAN) models. By introducing the attention mechanism, this model can distribute its attention to interactive information independently, which enhances the model's ability to predict multiple reasonable trajectories. Mohajerin and Rohani [35] employed an occupancy grid map to generate multistep predictions of a vehicle's drivable space. Here, by learning the differences between OGM sequences, this method provides accurate predictions for moving objects to effectively improve path planning and navigation. Choi [36] proposed a model to predict the trajectory of traffic agents in a highly interactive environment. The model can directly or indirectly affect the motion behavior of a target agent to correct the preliminary prediction results of that agent.

On urban roadways, various types of traffic agents coexist, e.g., cars, electric cars, and bicycles. However, all of the above methods predict the trajectory of a single agent; however, to the best of our knowledge, few studies have investigated the trajectory prediction of heterogeneous traffic agents. For example, Trafficpredict [37] constructed a 4-D graph. Here, two dimensions are used for agents and their interactions, where one dimension describes the temporal correlation, and another dimension considers agent classification. This method extracts both type features and individual motion features. Traphic [38] employed an LSTM-CNN hybrid network to predict the interaction between different road agents. This model implicitly considers the different motion patterns of different types of agents and to predict the trajectory of traffic agents in dense traffic videos. Zhao et al. [39] proposed a data-driven trajectory prediction model that takes into account the multimodality of the trajectory itself as well as group interactions, trained in an end-to-end fashion. In addition, Zhao et al. [40] proposed a vehicle and pedestrian trajectory prediction method based on deep learning, and they designed a multiagent tensor fusion network that can maintain spatial structure information to realize path planning that minimizes driving risks.

To solve the common limitations of the existing methods effectively, we propose the TraGCAN heterogeneous traffic agent trajectory prediction algorithm, which is based on a spatial attention network, for dense traffic scenarios. The proposed TraGCAN algorithm predicts the short-term movement trajectory in an end-to-end manner when the motion of vehicles, bicycles, and pedestrians affect each other.

III. Preliminary Knowledge

A. Problem Definition

In this study, we focus on the trajectory prediction of heterogeneous traffic agents in dense traffic, and this scenario primarily involves two aspects. First, we consider different types of traffic agents, e.g., pedestrians, bicycles, and vehicles. Second, dense traffic means that the distance between traffic agents is sufficiently close that the trajectory of each agent can be affected by neighboring agents. We give the following definitions to better describe the process.

Definition 1 (Target Traffic Agent): The traffic agent whose trajectory requires to be predicted.

Definition 2 (Surrounding Traffic Agents): Neighboring traffic agents on the road near the target traffic agent that affect the motional behavior of the target traffic agent.

The algorithm can predict the trajectory of the entire moving process of the target traffic agent accurately according to the historical states of the target traffic agent and surrounding traffic agents. Note that each traffic agent on the road can be considered the target traffic agent, and other traffic agents can be considered as surrounding traffic agents. Therefore, this model can predict the trajectories of all traffic agents on the road.

We describe the problem solved by TraGCAN as follows. Given a set of N traffic agents $A = \{a_i\}_{i=1\cdots N}$, the proposed algorithm analyzes the motion state of each agent in a frame-by-frame manner in the time interval $[1 : T_{\text{obs}}]$, including the traffic agent's position in the horizontal and vertical directions and other attributes. Then, the traffic agent's discrete position in each frame at $[T_{\text{obs}} + 1 : T_{\text{pred}}]$ is predicted.

B. Problem Analysis

The future trajectory of the traffic agent on the road is closely related to its own historical trajectory. Usually the motion state of agents does not change drastically in a short time, such as speed and direction, which is called temporal correlation. Meanwhile, the agents avoid collision and follow some social norms, so there are also complex spatial interactions between different traffic agents, which is called spatial correlation. We believe that the key to solve the trajectory prediction problem lies in accurately modeling spatial correlation and temporal correlation. In order to meet the needs of practical application scenarios, our proposed model focuses on both temporal and spatial aspects.

In Fig. 1(a), the three cars are driving to the right normally. When car B suddenly accelerates, the distance between B and A becomes closer, and the spatial relationship becomes stronger. This also causes the distance between B and C to become longer, and the space between B and C becomes longer.

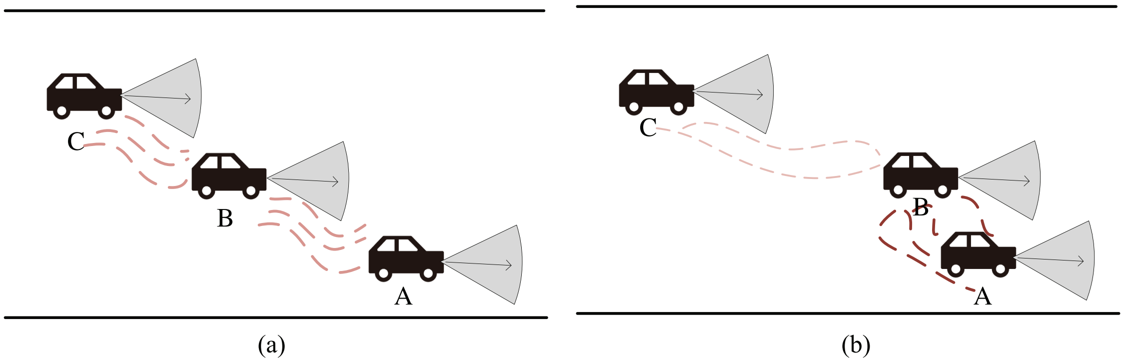


Fig. 1. Spatial relationship between traffic agents. Take the example of three cars A, B, and C driving on the road. The pink dotted line indicates the spatial relationship between the agents. The darker the color indicates a stronger spatial relationship (and vice versa). In general, the spatial relationship between traffic agents strongly related to the distance. (a) Original spatial relationship of traffic agents. (b) Adjusted spatial relationship of traffic agents.

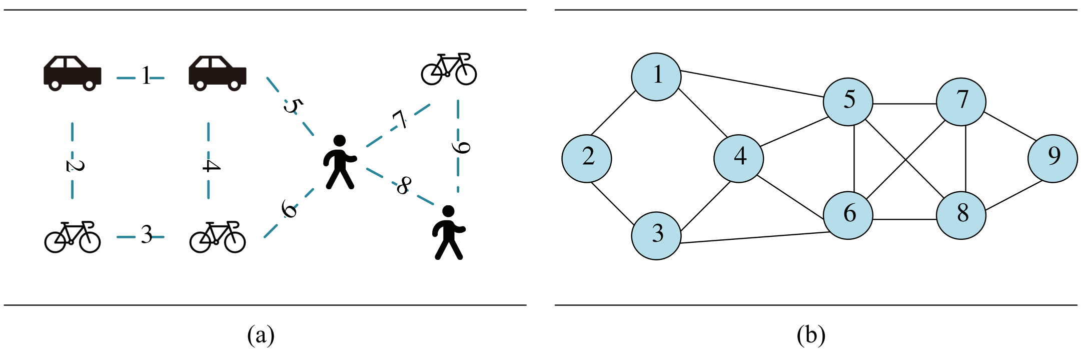


Fig. 2. (a) Figure shows a simple small network of all traffic agents which is then converted into a spatial graph according to the above procedure, and the conversion result is shown in (b).

The relationship becomes weaker, as shown in Fig. 1(b). We observe that spatial location changes of traffic agents cause changes in spatial relationships, and the interactions between neighboring traffic agents are more significant, and drivers tend to pay more attention to neighboring traffic agents in order to avoid collisions. The spatial relationships between agents also change over time due to the constant changes in the speed and direction of traffic agents on the road.

According to the above description, we can represent the spatial relationship between traffic agents in a graph structure. First, we believe that there is a spatial relationship between any two traffic agents on the road, from which we can build a preliminary diagram. Then, we extract the edges between nodes to build a spatial graph. In the spatial graph, we consider the spatial relationship between different agents as node V of the graph, and there is an edge E between two spatial relations with the same agent. Fig. 2(a) shows a simple small network of all traffic agents, which is then converted into a spatial graph according to the above procedure, and the conversion result is shown in Fig. 2(b).

IV. Overall Architecture

We analyze the trajectory data of all moving agents in space and time, and the overall structure of TraGCAN is shown in Fig. 3. The TraGCAN model comprises three main modules, i.e., the spatial relationship convolution, the spatial feature aggregation, and temporal feature aggregation modules. The spatial relationship convolution module mines neighbor information through the graph convolution network, which is used to capture the interaction between traffic agents, i.e., the spatial feature vector. The spatial feature aggregation module takes the spatial feature vector as input to calculate the degree of influence of the different movement modes of the surrounding agents on the movement of traffic agents. We then use the Attention LSTM to obtain the motion characteristics of the surrounding traffic agents. The temporal feature aggregation module takes the motion characteristics of the surrounding traffic agents and the state of the target traffic agents as input to the temporal LSTM to describe the change of trajectory of the traffic agents over time.

A. Input and Output

To clearly explain the process of the proposed algorithm, we first define the model input and output.

Definition 3 (Motion State): The motion state of each traffic agent a_i at any time t can be denoted as $f_i^t = (x_i^t, y_i^t, l_i, w_i, c_i)$, where x_i^t and y_i^t are coordinates in the lateral and longitudinal directions, respectively, at any time t. l_i and w_i represent the length and width of the traffic agent, respectively, and c_i is the type of traffic agent. $c_i \in \{1, 2, 3\}$, where 1 for pedestrians, 2 for bicycles, and 3 for vehicles.

Definition 4 (Spatial Feature): Considering the interaction between traffic agents, the spatial feature between agents a_i and a_j at time t is defined as $f_{ij}^t = (x_{ij}^t, y_{ij}^t, c_{ij})$.

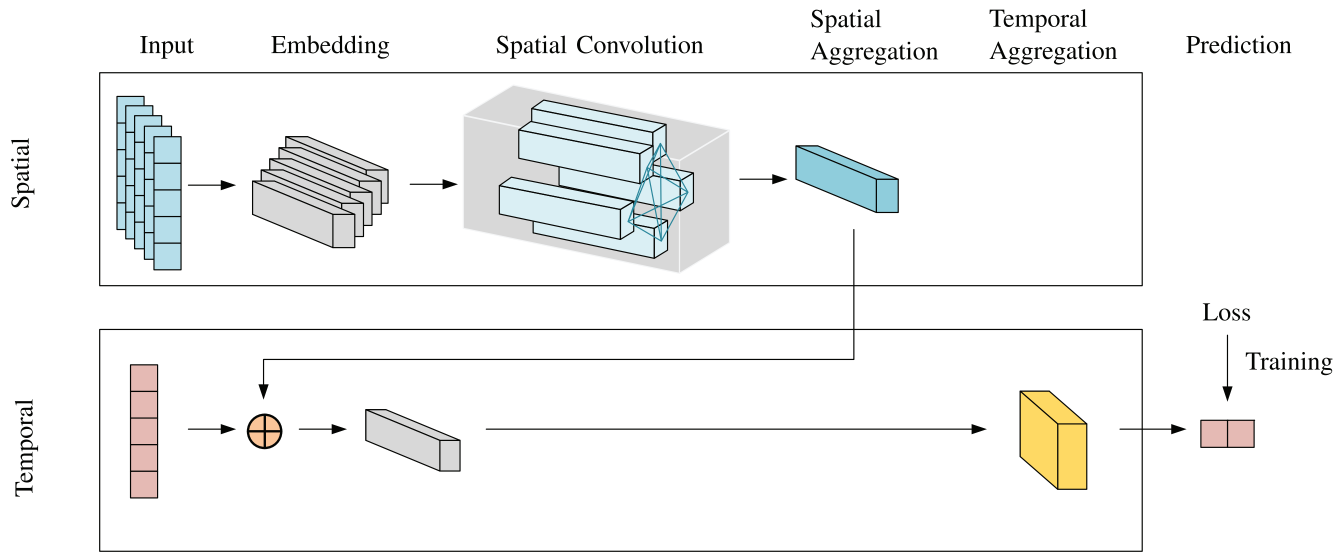


Fig. 3. Architecture of heterogeneous traffic trajectory prediction model based on graph convolution attention network. The model combines graph convolution to process trajectories from two dimensions, temporal and spatial, and finally outputs future trajectories.

$x_{ij}^t = x_j^t - x_i^t$, $y_{ij}^t = y_j^t - y_i^t$ represent the positional relationship between the two agents, respectively. Here, c_{ij} is the unique code corresponding to the motion state, and $c_{ij} = [c_i; c_j]$. This definition expresses the spatial correlation of the two traffic agents to a certain extent.

In the previous definitions, f_i^t and f_{ij}^t contain the position, displacement, size, and type of the traffic agent at any given time. The speed of a traffic agent can be represented by its displacement in adjacent frames. The greater the displacement, the faster the speed. Note that there are obvious differences in turning radius between traffic agents of different sizes and types, and an agent's speed is closely related to the turning radius, i.e., faster the speed, the larger the turning radius. For example, if the vehicle is not turning at an appropriate speed, the driver may lose control, e.g., a rollover may occur. Therefore, our definition implicitly considers two factors, i.e., the speed and turning radius of traffic agents.

We take the motion state sequence $F_i^{T_{\text{obs}}} = [f_i^1, f_i^2, \ldots, f_i^{T_{\text{obs}}}]$ of each target traffic agent a_i and the spatial feature sequences $F_{ij}^{T_{\text{obs}}} = [f_{ij}^1, f_{ij}^2, \ldots, f_{ij}^{T_{\text{obs}}}]$ between the a_i and the surrounding traffic agents as the inputs of the model. The final output of the model is the predictive position sequence $\text{Tr}_i^{T_{\text{pred}}} = [\text{tr}^{T_{\text{obs}}+1}, \text{tr}_i^{T_{\text{obs}}+1}, \ldots, \text{tr}_i^{T_{\text{pred}}}]$ of the target traffic agent a_i. The position $\text{tr}_i^{t'}$ of a_i at time t' is denoted as $(x_i^{t'}, y_i^{t'})$.

B. Probabilistic Trajectory Prediction

Our position prediction method is the same as the existing S-LSTM method [25]. Here, we assume that the position of the traffic agent at the next time follows a bivariate Gaussian distribution, including parameter expectation $\mu_i^{t+1} = (\mu_x, \mu_y)_i^{t+1}$, standard deviation $\sigma_i^{t+1} = (\sigma_x, \sigma_y)_i^{t+1}$, and correlation coefficient ρ_i^{t+1}. The position of the traffic agent at time $t+1$ is expressed as follows:

$$\left(x_i^{t+1}, y_i^{t+1}\right) \sim N\left(\mu_i^{t+1}, \sigma_i^{t+1}, \rho_i^{t+1}\right). \quad (1)$$

We train the model by minimizing the negative log-likelihood loss function of the ith traffic agent. The loss function is expressed as follows:

$$L_i = -\sum_{t=T_{\text{obs}}}^{T_{\text{pred}}-1} \log\left(P\left(x_i^{t+1}, y_i^{t+1} | \mu_i^{t+1}, \sigma_i^{t+1}, \rho_i^{t+1}\right)\right). \quad (2)$$

We jointly backpropagate through the entire network to optimize the TraGCAN parameters in order to minimize the position error at each time step.

V. Algorithm Description

A. Spatial Relationship Convolution Module

The trajectories of the agents also change over time due to the interaction between the traffic agents on the road. Therefore, accurate modeling of complex spatial interactions between traffic agents is crucial for the accurate prediction of future trajectories. In this work, we introduce a graph convolution algorithm, which is used to extract complex spatial interaction features between traffic agents (Fig. 4).

Given a spatial relationship connectivity graph $G = (V, E)$, we set the information carried by each node in the graph G as the spatial feature f_{ij}^t between the traffic agents, which is also the input of this module. Here, we use the embedding layer to convert f_{ij}^t to a fixed-size vector

$$e_{ij}^t = \phi\left(W_{\text{spa}} f_{ij}^t + b_{\text{spa}}\right) \quad (3)$$

where W_{spa} and b_{spa} represent the weight matrix and bias vector, respectively, and $\phi(\cdot)$ is a nonlinear activation function.

We take the reciprocal of the sum of the distance between the nodes connected by two spatial relationships as the edge weight $a_{ij}^{(t)}$, which can be combined into a graph-weighted adjacency matrix A_t. The topological structure of the graph does not change with time. The weight of the edge will change due to the continuous movement of the node position. To facilitate learning, we first standardize the adjacency

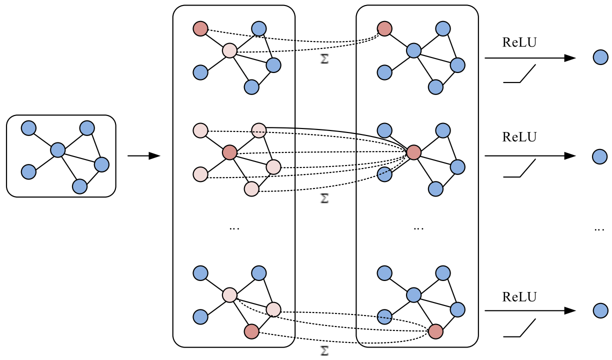


Fig. 4. Graph convolutional network based on a spatially connectivity graph, where nodes aggregate information from their neighbors and update their own features. The graph convolutional network is well suited to handle complex spatial interactions among traffic agents in real road scenarios.

matrix, which is very important to realize graph convolution, as described in the literature. The adjacency matrix is expressed as $A = \{A_{t-t_{\text{obs}}}, \ldots, A_{t-1}, A_t\}$, and we use the following formula to complete the standardization operation:

$$A_t = \Lambda_t^{-\frac{1}{2}} \hat{A}_t \Lambda_t^{-\frac{1}{2}}. \tag{4}$$

Next, we implement the graph convolutional network to complete extraction of the spatial interaction feature vector h_{gcn}^t

$$H_{\text{gcn}}^t = \sigma\left(\Lambda^{-\frac{1}{2}} \hat{A} \Lambda^{-\frac{1}{2}} E^t W_{\text{gcn}}\right). \tag{5}$$

Here, E^t is the vector matrix composed of e_{ij}^t. The output vector $H_{\text{gcn}}^t = \{h_{ij}^t | i \neq j\}$ is a potential representation that contains spatially useful information. h_{ij}^t describes the interaction between the target traffic agent and the surrounding traffic agents. W_{gcn} represents the weight matrix.

B. Spatial Feature Aggregation Module

The trajectory of the target moving agent is affected by the surrounding moving agents, so different spatial interaction feature vectors must be aggregated. In addition, for each agent, due to different factors, such as location and speed, the degree of influence of other agents on the target mobile agent is also different. Therefore, a new attention mechanism is introduced to adaptively calculate the weights of all spatial interaction features, selectively focus on some important features while selectively ignore other features, as shown in Fig. 5.

The current attention mechanism is mostly a static structure. The calculation of attention score pays attention to the state information at the current moment, only considers the time change from one level, and ignores the difference of influencing factors. We design a dynamic attention mechanism. In this mechanism, we introduce a global feature that changes over time. It will ensure that the results of the attention score calculation are more efficient. We capture the dynamic spatial correlation between different traffic agents at time t through the attention mechanism

$$o_{ij}^t = V_l \tanh\left(W_l d_i^t + U_l h_{ij}^t + b_l\right) \tag{6}$$

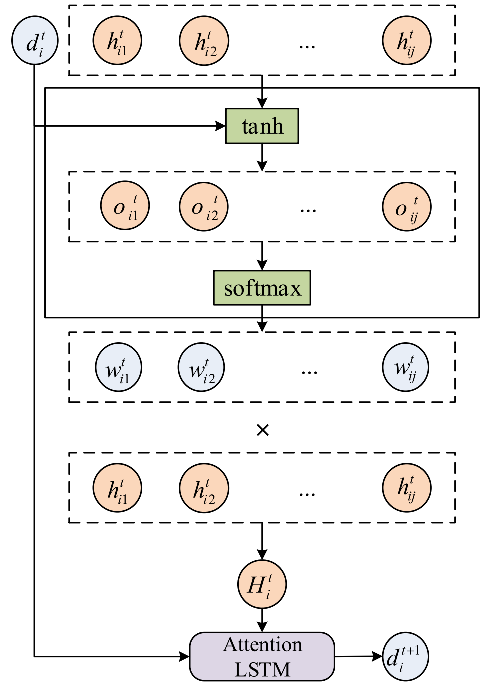


Fig. 5. Dynamic attention mechanism. We introduce a global feature d_i^t that changes over time. It will ensure that the results of the attention score calculation are more efficient.

$$w_{ij}^t = \text{softmax}\left(o_{ij}^t\right) \tag{7}$$

$$H_i^t = w_{i1}^t h_{i1}^t + w_{i2}^t h_{i2}^t + \cdots + w_{ij}^t h_{ij}^t \tag{8}$$

where d_i^t represents the movement features of surrounding agents. h_{ij}^t is obtained from Spatial LSTM. V_l, W_l, and U_l are weight matrix, b_l is the bias vector, and w_{ij}^t represents the influence of the movement state of agent a_j on the trajectory of agent a_i, i.e., the interaction effects between agents. Then, the hidden state vectors of all agents around a_i are added based on different interaction effects. As a result, we obtain H_i^t to evaluate the spatial interaction between a_i and the surrounding agents, i.e., the comprehensive impact of all other agents on target agent a_i.

H_i^t is the input of Attention LSTM, which is used to update the motion features d_i^{t+1} of the surrounding agents

$$d_i^{t+1} = \text{LSTM}\left(H_i^t, d_i^t, W_{\text{attn}}, b_{\text{attn}}\right) \tag{9}$$

where W_{attn} and b_{attn} represent the weight matrix and bias vector of Attention LSTM, respectively. Here, d_i^{t+1} obtained by Attention LSTM integrates the influence of the movement of all surrounding traffic agents on the future trajectory of the target traffic agent, which represents the comprehensive motion feature of the surrounding agents.

C. Temporal Feature Aggregation Module

Different traffic agents exhibit different motion patterns, e.g., the agent position of a vehicle changes constantly, and the speed will also change. Note that the trajectory of the traffic agent is time dependent; thus, we introduce Temporal LSTM to extract the motion feature of the target traffic agent over time.

TABLE I
EXAMPLE OF THE DETAILS OF EACH FIELD IN BAIDU'S APOLLOSCAPE AUTONOMOUS DRIVING DATA SET. THESE FIELDS CONSIST OF FRAME ID, OBJECT ID, OBJECT TYPE (DIFFERENT TYPES OF AGENTS ARE REPRESENTED BY DIFFERENT NUMBERS), POSITION, SIZE, AND HEADING ATTRIBUTES

frame_id	object_id	object_type	position_x	position_y	position_z	object_length	object_width	object_height	heading
0	1	2	119.459	64.591	39.448	11.101	3.134	3.276	-3.116
0	2	3	111.692	82.858	38.785	0.348	0.594	1.168	0.03
1	1	2	125.377	65.911	39.337	10.953	3.029	2.923	-3.121
1	2	3	110.794	82.738	38.811	0.479	0.522	1.323	0.046

The behavior of traffic agents is influenced by other agents; thus, we obtain an embedded vector e_i^t through the motion state f_i^t of the target traffic agent a_i and the output d_i^t of Attention LSTM as the input of Temporal LSTM

$$e_i^t = \phi\left(W_{\text{tem1}}\left[f_i^t; d_i^t\right] + b_{\text{tem1}}\right) \tag{10}$$

where W_{tem1} and b_{tem1} are weight matrix and bias vector, respectively. The speed and turning radius also affect the behavior of traffic agents. For example, because pedestrians and bicycles move slowly, they can change the driving direction and change the trajectory in a short time to avoid collision events, which is different from the interaction between vehicles and other agents

$$h_i^{t+1} = \text{LSTM}\left(e_i^t, h_i^t, W_{\text{tem2}}, b_{\text{tem2}}\right) \tag{11}$$

here, W_{tem2} and b_{tem2} represent the weight matrix and bias vector of Temporal LSTM, respectively. The motion feature h_i^{t+1} of the traffic agent is the output of Temporal LSTM. h_i^{t+1} represents the movement features of the target traffic agent over time, which is the key feature to predict the position at the next time.

These parameters required for trajectory probability prediction are determined by the hidden vector h_i^{t+1} at time $t+1$

$$\left[\mu_i^{t+1}, \sigma_i^{t+1}, \rho_i^{t+1}\right] = W_p h_i^t + b_p. \tag{12}$$

W_p and b_p represent the weight matrix and bias vector, respectively. The predicted position of the traffic agent at time $t+1$ can be derived from these.

VI. EXPERIMENT

A. Data Set

We first used the Apolloscape autonomous driving data set[1] provided by Baidu to train and evaluate the proposed model. This data set contains the trajectory information of traffic agents under different lighting conditions and traffic densities in Beijing, China captured using multiple sensors, e.g., radar devices, and GPS. The traffic agents are include different types of pedestrians, bicycles, and vehicles. This data set contains a 53-min training sequence and a 50-min test sequence. The sampling frequency is two frames per second. The information of each agent includes the agent ID, type, position, size, and deflection angle. Table I shows examples from the data set. Note that the Apolloscape autonomous driving data set conforms to the target dense traffic scenarios.

[1]Apolloscape, Trajectory data set for urban traffic, 2018. http://apolloscape.auto/trajectory.html.

B. Experimental Setup

After many experiments and comparison of different parameters, we employed an embedding layer with 64 dimensions to input the spatial coordinates into a single layer LSTM, and we set the hidden state dimension of all LSTM models to 128. To achieve the optimal effect of the model, we employed the Adam optimizer to train the entire network with a batch size of 8 and a learning rate of 0.001. In addition, weight decay was used to reduce overfitting. Considering the real-time changes of the traffic agent trajectory, the autonomous driving trajectory prediction algorithm pays more attention to the position movement of the agents in the short term. We used the historical motion state of the traffic agent in the past 2 s to predict its coordinates in the lateral and longitudinal directions in the next 3 s. In addition, we normalized the data set to improve the accuracy of the model. The proposed model was implemented in Pytorch using a single GPU.

C. Evaluation Metrics and Compared Methods

We considered the following common metrics to evaluate the performance of different traffic agent trajectory prediction algorithms [25], [38], [41]. To compare a distribution with a certain target value, we follow the evaluation method used in S-LSTM [25] in which 20 samples are generated based on the predicted distribution. The models used the best amongst 20 samples for evaluation.

1) *Average Displacement Error (ADE):* Average L2 distance between ground truth and our prediction overall predicted time steps

$$\text{ADE} = \frac{\sum_{n\in N}\sum_{t\in[T_{ob+1},T_{\text{pred}}]}\left\|\hat{y}_{\text{t}}^n - y_{\text{t}}^n\right\|_2}{N\times\left(T_{\text{pred}} - T_{ob}\right)}. \tag{13}$$

2) *Final Displacement Error (FDE):* The distance between ground truth and the predicted final destination at end of the prediction period T_{pred}

$$\text{FDE} = \frac{\sum_{n\in N}\left\|\hat{y}_t^n - y_t^n\right\|_2}{N}, t = T_{\text{pred}}. \tag{14}$$

The proposed TraGCAN algorithm was compared to the following methods.

S-LSTM: The S-LSTM algorithm is based on LSTM that includes a social pooling layer to integrate the hidden state of other pedestrians and predict the trajectory of pedestrians in traffic [25].

S-GAN: The Social-GAN algorithm is based on sequence prediction and generates an adversarial network to predict pedestrian movement behavior [42].

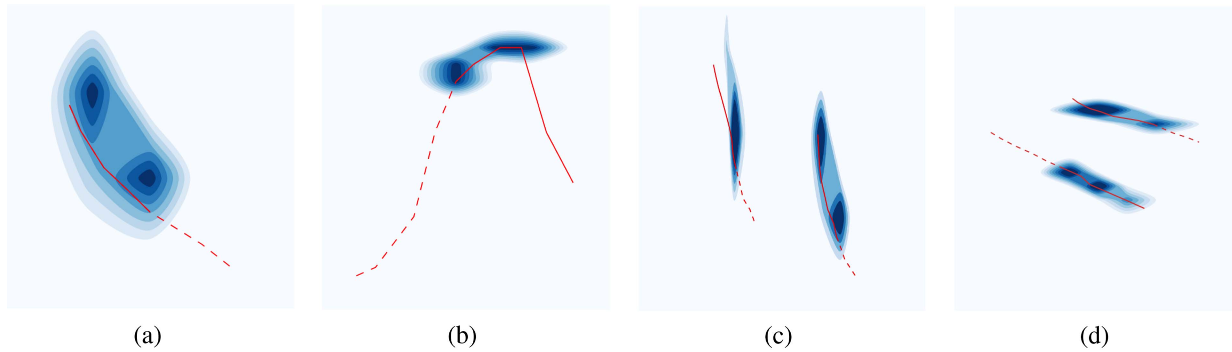


Fig. 6. Qualitative analysis of TraGCAN. The trajectory prediction results for four different road scenarios are shown here. The red dotted line part represents the historical trajectory of the traffic agent, the red solid line represents the ground truth, and blue areas represent the probability distribution of trajectories predicted by our GraGCAN algorithm. The darker the color, the larger the probability distribution of the location. (a) Agent moving on straight lanes. (b) Agent moving along the curve. (c) Two agents meeting from the same direction. (d) Two agents meeting from different directions.

TrafficPredict: This LSTM-based algorithm uses the structure of a 4-D graph to predict the movement trajectory of different types of traffic agents [37].

S-STGCNN: S-STGCNN models uses graph convolutional network instead of interactive aggregation mechanism, and uses TCN instead of recursive architecture [31].

D. Analysis of Results

Fig. 6 shows the trajectory prediction results for four different motion states, i.e., a single agent going straight and turning, and multiple agents in the same direction and reverse. In Fig. 6, the red dotted line part represents the historical trajectory of the traffic agent, the red solid line represents the ground truth, and blue represents the probability distribution of trajectories predicted by the model. The darker the color, the larger the probability distribution. As can be seen, the model can obtain a reasonable direction of movement for different agent motion states.

We compared the proposed TraGCAN algorithm to the existing methods on the Apolloscape data set and recorded the ADE and FDE of three different traffic agents (pedestrian, bicycle, and vehicle agents).

Table II shows the best performance of each method under different parameter settings. As can be seen, the proposed TraGCAN outperformed all compared methods, and the error was reduced by 15% compared to the TrafficPredict algorithm. The S-LSTM and S-GAN are two common trajectory prediction methods; however, their target objects are pedestrians. There are large differences in the movement behavior between pedestrians and vehicles; thus, so they cannot be well adapted to heterogeneous traffic on dense roads and cause large errors. Social STGCNN employs the graph convolution technology to explore the influence between adjacent pedestrians, which can simulate real interactions between pedestrians to a certain extent. The overall performance was better than S-LSTM and S-GAN. Traffic Predict considers the interaction between heterogeneous agents in space and time in its model, and the error of prediction results is reduced. However, the result of this method was not optimal because it relies on the position of the traffic agents as the input to the model. The proposed TraGCAN algorithm considers both the difference in the degree of interaction between heterogeneous agents and the hidden characteristics of each agent. In addition, it makes fully exploits the state information of the traffic agent; thus, it demonstrated excellent performance on the experimental data set.

TABLE II
COMPARISON OF EXPERIMENTAL RESULTS OF DIFFERENT TRAJECTORY PREDICTION METHODS. THE TABLE LISTS THE PREDICTION ERRORS AND AVERAGE ERRORS OF DIFFERENT ALGORITHMS FOR DIFFERENT TYPES OF TRAFFIC AGENTS. THE MODELS USED THE BEST AMONGST 20 SAMPLES FOR EVALUATION. ADE AND FDE ARE IN METERS, THE LOWER THE BETTER. THE ADE AND FDE VALUES OF THE PREDICTED RESULTS WERE CALCULATED ACCORDING TO THREE TYPES OF TRAFFIC AGENTS (PEDESTRIANS, BICYCLES, AND CARS). VALUES IN BOLD REPRESENT THE OPTIMAL RESULTS

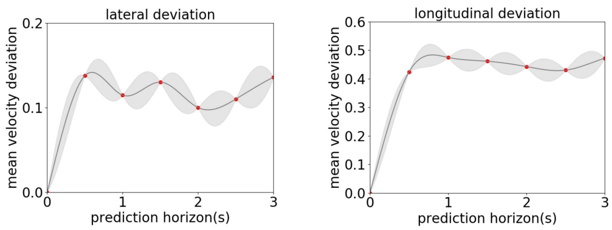
ADE Metric					
Agent type	S-LSTM	S-Gan	Traffic Predict	S-STGCNN	TraGCAN
pedestrian	0.128	0.124	0.089	0.116	**0.069**
bicycle	0.145	0.138	0.096	0.135	**0.091**
vehicle	0.147	0.141	0.138	0.121	**0.100**
average	0.140	0.134	0.109	0.124	**0.087**

FDE Metric					
Agent type	S-LSTM	S-Gan	Traffic Predict	S-STGCNN	TraGCAN
pedestrian	0.193	0.205	0.127	0.152	**0.094**
bicycle	0.235	0.21	0.133	0.177	**0.126**
vehicle	0.228	0.213	0.189	0.168	**0.145**
average	0.219	0.209	0.150	0.166	**0.122**

We compared the above trajectory prediction methods from different perspectives, including scenario, traffic agent type, traffic agent interaction, and other constraints, to clearly distinguish the different problems that different methods are adapted to. Table III shows that the proposed TraGCAN algorithm is suitable for solving urban traffic scenarios that involve interactions between the same and different types of traffic agents. The TraGCAN algorithm implicitly introduces the movement speed and turning radius of traffic agents considers various factors more comprehensively, and provides interpretability in terms of its low error prediction results. The three compared methods consider factors different from that of the proposed TraGCAN algorithm. For example, S-LSTM, S-GAN, and S-STGCNN are suitable for single-agent trajectory prediction. In addition, TrafficPredict considers heterogeneous traffic; however, it does not analyze the characteristics of different agents, which explains why

TABLE III
MULTIDIMENSIONAL COMPARISON OF DIFFERENT TRAJECTORY PREDICTION METHODS

Method	Scenario	Traffic agent type	Traffic agent interaction	Other constraints
S-LSTM	Crowded space	Pedestrians	Pedestrian interaction	No
S-GAN			Pedestrian interaction and trajectory information sharing	
S-STGCAN				
Traffic Predict	Heterogeneous traffic agents in urban roads	Pedestrians, bicycles, vehicles	There are interactions between the same and different traffic agents	
TraGCAN			There are interactions between the same and different types of traffic agents, and their interactions are different	Introduce the position displacement and size of the traffic agent, and implicitly consider its moving speed and turning radius



(a) (b)

Fig. 7. Average velocity error of predicted trajectory (km/h): Fig. 7(a) and (b) show the change of traffic agent velocity along the lateral and longitudinal directions with the predicted time, respectively, where velocity is defined as the position change of adjacent time intervals. (a) Lateral. (b) Longitudinal.

this method performance slightly worse than the proposed TraGCAN algorithm.

For the proposed TraGCAN algorithm, we calculated the velocity deviations in different prediction time ranges to analyze the main causes of trajectory prediction errors (Fig. 7). Since the speeds of different types of agents are significantly different, we only choose the trajectories of some vehicles traveling in the same direction for analysis. The velocity is the position change of the agents in the lateral and longitudinal directions in adjacent time intervals. In Fig. 7, x-axis represents different forecast time frames (0.5 s, 1 s, 1.5 s, 2 s, 2.5 s, and 3 s), and the y-axis is the velocity deviation between the predicted results and the true results. Fig. 7(a) and (b) shows that, although the distribution of velocity errors in the lateral and longitudinal directions in different time ranges is approximately the same, the accumulation of velocity errors increases the overall prediction error. The moving velocity of pedestrians is much smaller than that of vehicles; thus, the prediction errors caused by velocity errors are smaller, and the accuracy of the prediction results is higher.

To demonstrate the trajectory prediction results more intuitively, we randomly selected the trajectories of three different agents and mapped them in a 2-D coordinate space to observe the position change of the agents in a future period (Fig. 8). Fig. 8(a) shows that when the traffic agent moves along a straight line, the predicted trajectories of all methods were close to the real trajectory on the ground with high accuracy. Therefore, when the motion state of the target agent is relatively stable, the predicted results of the model are more accurate.

The algorithm can predict the deflection trend of a traffic agent when its direction deflects in the observation. However, when the agent's direction suddenly deflects during inference, the algorithm may not fully capture the deflection signal, resulting in errors in the predicted trajectory. Compared with existing methods, the proposed TraGCAN algorithm exhibits minimal deviation from the ground truth in all cases. Note that S-GAN did not predict the deflection trend of the agent, and the result deviated significantly from the ground truth. Other methods can predict the deflection trend of the agent; however, there is still a certain deviation from the ground truth.

Then, we selected three common road scenarios, namely, lane change, crossroads, and collision avoidance and presented the visualized prediction results in Fig. 9. These three realistic traffic scenarios with heterogeneous traffic agents pose challenges for trajectory prediction. We compare the proposed algorithm TraGCAN with the corresponding baselines in these three realistic traffic scenarios. We observed that our proposed algorithm is able to sensitively capture small changes in future trajectories due to traffic agent direction and speed changes, as confirmed by the lane change and collision avoidance plots located in the upper right corner. The visualization results demonstrate that the predictions of all the algorithms do not deviate too much from the ground truth in the short term. The prediction results of our algorithms are more accurate in most scenarios. But there is no magic in the world, as we show an example of a failed prediction in the lower right corner.

VII. ABLATION STUDY AND QUALITATIVE ANALYSIS

A. Does TraGCAN Really Work?

The proposed TraGCAN algorithm differs from existing algorithms in several ways. For example, in the TraGCAN model, we consider the size and type of traffic agents, and we implicitly introduce factors, such as speed and turning radius of the agents, which are the inputs to the algorithm. In addition, the proposed algorithm employs an attention mechanism to aggregate the effects of different spatial interactions on the target traffic agents. The excellent performance of the TraGCAN algorithm is due to its consideration of these many different factors. To further verify the effectiveness of these factors, we conducted experimental comparisons of the proposed algorithm with different settings.

TraGCAN-nic: In this case, the proposed algorithm did not consider the agent constraints, e.g., size and type.

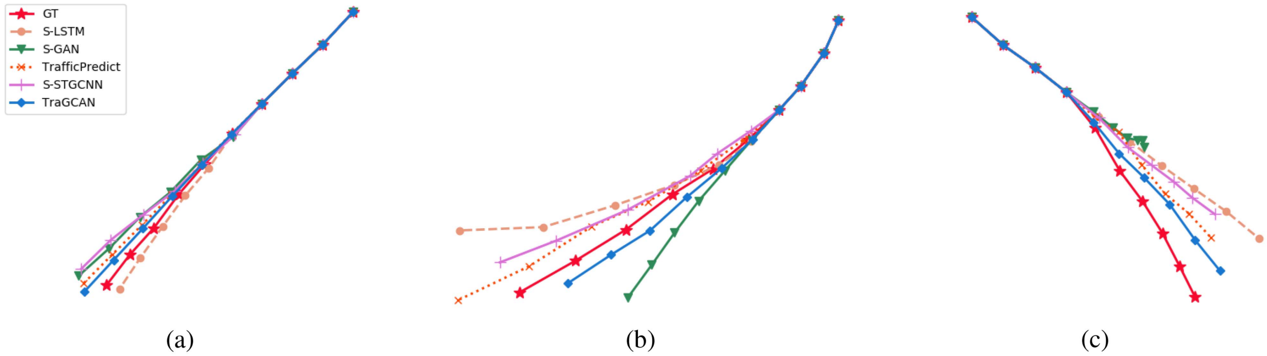


Fig. 8. Comparison of predicted trajectory to the real trajectory on the ground. Fig. 8(a)–(c) illustrate the trajectory prediction results of different algorithms for three traffic agents. For each trajectory, the first four nodes are the historical positions of the observations. The last six nodes are the prediction results. The ground true trajectory is shown in red, and the prediction result of the proposed TraGCAN algorithm is shown in blue.

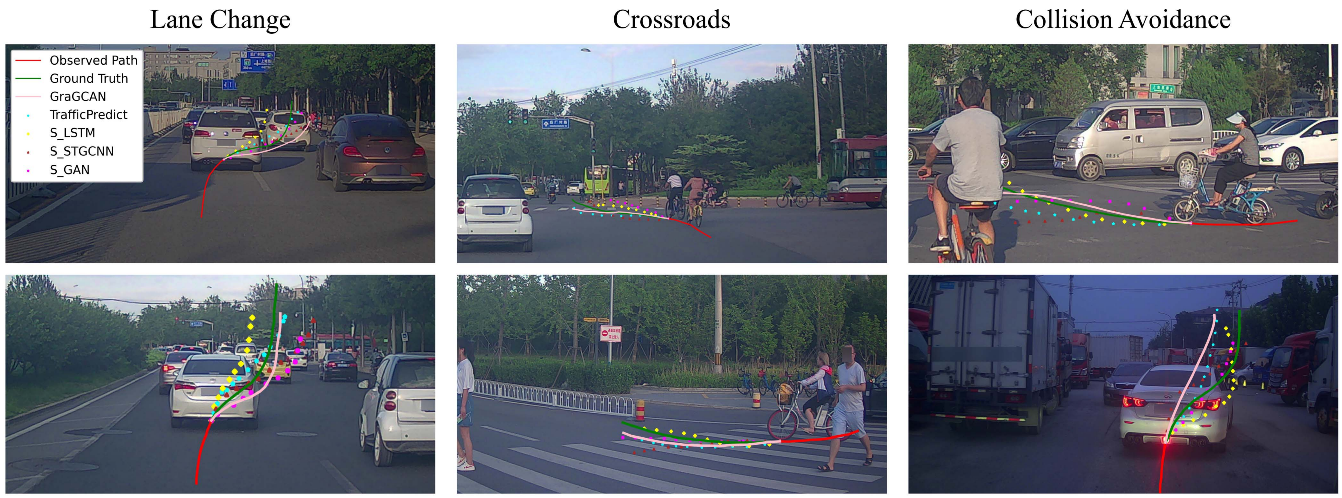


Fig. 9. Illustration of our algorithm TraGCAN compared with other extant algorithms on the Baidu Apolloscape data set. We have selected three common road scenarios, namely, lane change, crossroads, and collision avoidance, and each column represents a scenario. The observed trajectories and ground truth are drawn with red and green lines, respectively. The predicted trajectories of our algorithm TraGCAN are drawn with pink lines, and the predicted trajectories of other algorithms are depicted with dots of different colors. The visualization results show that our algorithm predicts more accurately in most realistic traffic scenarios.

TABLE IV
COMPARISON OF EXPERIMENTAL RESULTS WITH DIFFERENT TRAGCAN SETTINGS. VALUES IN BOLD REPRESENT THE OPTIMAL RESULTS. ADE AND FDE ARE IN METERS, THE LOWER THE BETTER

ADE Metric			
Agent type	TraGCAN-nic	TraGCAN-nsa	TraGCAN
pedestrian	0.079	0.119	**0.069**
bicycle	0.092	0.134	**0.091**
vehicle	0.115	0.14	**0.100**
average	0.095	0.131	**0.087**

FDE Metric			
Agent type	TraGCAN-nic	TraGCAN-nsa	TraGCAN
pedestrian	0.106	0.176	**0.094**
bicycle	0.138	0.198	**0.126**
vehicle	0.161	0.201	**0.145**
average	0.135	0.191	**0.122**

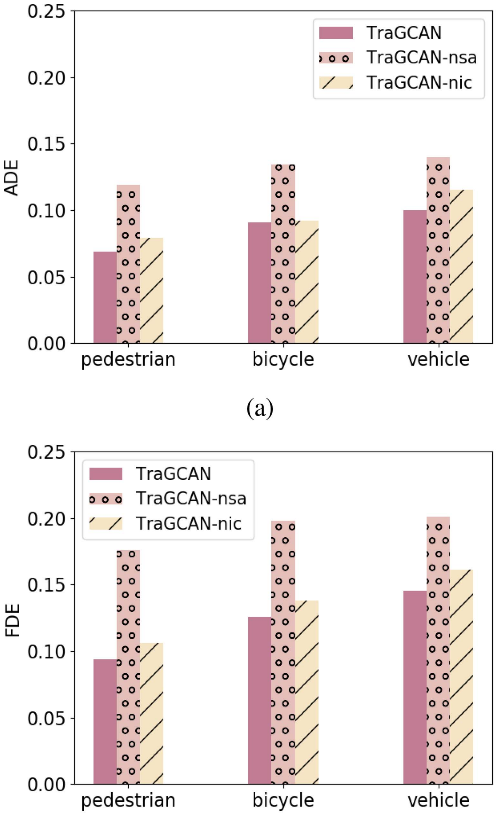
TraGCAN-nsa: In this case, the proposed algorithm did not use the attention mechanism.

The experimental results are presented in Table IV. We found that TraGCAN-nsa distinguished different types of traffic agents and integrated their dynamic constraints. However, as all traffic agents in this method have the same influence on the target agents, the prediction error of the final model is large. By including the attention mechanism, TraGCAN-nic adaptively captured different degrees of spatial influence, and the prediction results were improved significantly. The proposed TraGCAN algorithm represents a combination of the above two methods, which considers more comprehensive factors and has better performance.

Fig. 10 shows that TraGCAN-nsa has the highest error of the three methods, suggesting that the dynamic attention mechanism plays a key role in accurately predicting the trajectory of heterogeneous entities.

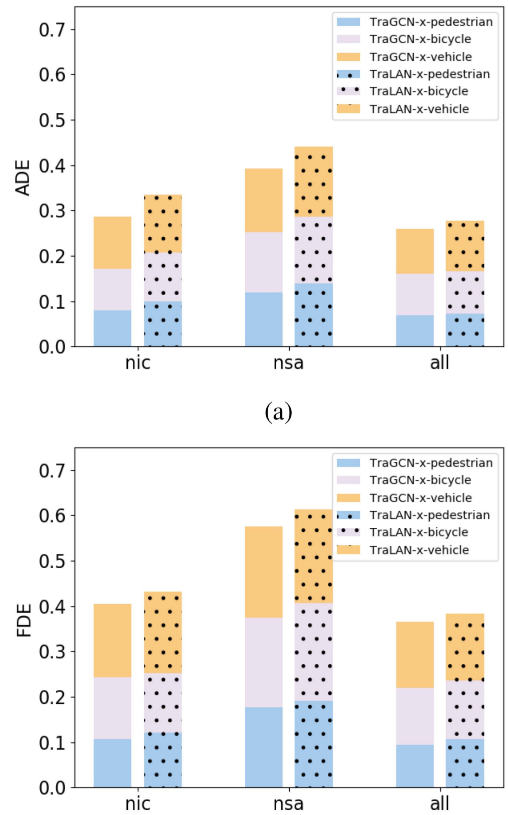
B. GCN Versus LSTM?

The proposed algorithm innovatively uses graph convolutional networks to handle the feature extraction problem because the graph convolution method can extract spatial features from the topology map effectively, and accurately



(a) (b)

Fig. 10. Comparison of experimental results in TraGCAN, TraGCAN-nsa, and TraGCAN-nic. The comparison results of ADE and FDE are shown in Fig. 10(a) and (b), respectively. ADE and FDE are in meters, the lower the better. (a) ADE of TraGCAN, TraGCAN-nsa, and TraGCAN-nic. (b) FDE of TraGCAN, TraGCAN-nsa, and TraGCAN-nic.



(a) (b)

Fig. 11. Comparison of experimental results based on GCN and LSTM models. ADE and FDE are in meters, the lower the better. (a) ADE for GCN versus LSTM. (b) FDE for GCN versus LSTM.

extracting the spatial interaction features between traffic agents is a key step in the process of trajectory prediction.

From previous comparison experiments, we found that the TraGCAN model that includes a graph convolution module achieves smaller prediction errors. To further explore the role of GCN in the model and to verify the effectiveness of GCN, we compared GCN-based and LSTM-based algorithms and tested the effectiveness of these algorithms in extracting spatial interaction features. Next, we briefly describe the LSTM-based trajectory prediction model (TraLAN).

The spatial correlation between traffic agents changes over time due to changes in the agents' states. For example, if two adjacent vehicles traveling at a slow speed increase their speed at the next moment, their spatial correlation will become stronger as the speed increases. To this end, we consider the spatial interaction between different traffic agents as a time series that can be processed by the LSTM model; thus, changes in the interaction effect over time can be captured.

Here, we define an embedding vector e_{ij}^t based on the motion status of surrounding traffic agents f_{ij}^t as follows:

$$e_{ij}^t = \phi\left(W_{\text{spa1}} f_{ij}^t + b_{\text{spa1}}\right) \tag{15}$$

where W_{spa1} and b_{spa1} represent the weight matrix and bias vector, respectively, and $\phi(\cdot)$ is a nonlinear activation function. We then use embedding vector e_{ij}^t as the input of Spatial LSTM to update the hidden state h_{ij}^t of LSTM

$$h_{ij}^t = \text{LSTM}\left(e_{ij}^t, h_{ij}^{t-1}, W_{\text{spa2}}, b_{\text{spa2}}\right) \tag{16}$$

where the output vector h_{ij}^t of Spatial LSTM is a potential representation containing spatially useful information. h_{ij}^t can describe the interaction between the target traffic agent and the surrounding traffic agents over time. W_{spa2} and b_{spa2} represent the weight matrix and bias vector of Spatial LSTM, respectively.

After the spatial interaction features are obtained, the same spatial feature aggregation and time feature aggregation modules as TraGCAN are used to complete the final trajectory prediction. In order to make the test results fair, we made changes to TraLAN and obtained TraLAN-nic and TraLAN-nsar to compare it with our model in depth. The experimental results are shown in Table V and Fig. 11. From the data, It is easy to conclude that the prediction error of the proposed TraGCAN algorithm is less than that of TraLAN, which indicates that the GCN demonstrates better spatial interactive feature extraction performance.

TABLE V
COMPARISON OF TRALAN AND TRAGCAN UNDER DIFFERENT SETTINGS IS SHOWN. THE TABLE LISTS THE PREDICTION ERRORS AND AVERAGE ERRORS OF DIFFERENT ALGORITHMS FOR DIFFERENT TYPES OF TRAFFIC AGENTS. ADE AND FDE ARE IN METERS, THE LOWER THE BETTER. VALUES IN BOLD ARE THE OPTIMAL RESULTS

ADE Metric						
Agent type	TraGCAN-nic	TraGCAN-nsa	TraGCAN	TraLAN-nic	TraLAN-nsa	TraLAN
pedestrian	0.079	0.119	**0.069**	0.099	0.138	0.073
bicycle	0.092	0.134	**0.091**	0.109	0.149	0.092
vehicle	0.115	0.14	**0.100**	0.127	0.154	0.112
average	0.095	0.131	**0.087**	0.112	0.147	0.092

FDE Metric						
Agent type	TraGCAN-nic	TraGCAN-nsa	TraGCAN	TraLAN-nic	TraLAN-nsa	TraLAN
pedestrian	0.106	0.176	**0.094**	0.121	0.191	0.107
bicycle	0.138	0.198	**0.126**	0.132	0.215	0.129
vehicle	0.161	0.201	**0.145**	0.179	0.207	0.148
average	0.135	0.191	**0.122**	0.144	0.204	0.128

TABLE VI
EXPERIMENTAL RESULTS OBTAINED ON NGSIM I-80 AND US-101 DATA SETS. ADE/FDE METRICS FOR SEVERAL METHODS COMPARED TO TRAGCAN ARE SHOWN. ADE AND FDE ARE IN METERS, THE LOWER THE BETTER. VALUES IN BOLD ARE THE OPTIMAL RESULTS

ADE Metric							
NGSIM Dataset	S-LSTM	S-Gan	Traffic Predict	S-STGCNN	TraGCAN-nic	TraGCAN-nsa	TraGCAN
I-80	0.307	0.271	**0.263**	0.306	0.271	0.284	0.283
US-101	0.315	0.305	0.288	0.297	**0.278**	0.29	0.288
average	0.311	0.288	**0.275**	0.301	**0.275**	0.287	0.286

FDE Metric							
NGSIM Dataset	S-LSTM	S-Gan	Traffic Predict	S-STGCNN	TraGCAN-nic	TraGCAN-nsa	TraGCAN
I-80	0.544	0.572	**0.533**	0.581	0.549	0.568	0.556
US-101	0.542	0.593	0.562	0.578	**0.53**	0.562	0.578
average	0.543	0.582	0.547	0.580	**0.54**	0.565	0.567

C. Does TraGCAN Still Works Under Different Data Sets?

To verify the versatility of the proposed model, we conducted experiments on the NGSIM I-80 and US-101 data sets.[2] The I-80 data set includes vehicle trajectory data collected on the I-80 eastbound highway in Emeryville, California on 13 April 2005. The study area is approximately 500 m in length and includes six highway lanes. The US-101 data set includes vehicle trajectory data collected on the US-101 southbound highway in Los Angeles, California on 15 June 2005. The study area is approximately 640 m in length and includes five major traffic lines. The NGSIM data set can be used to study trajectory planning in highway scenarios.

The proposed TraGCAN algorithm was designed to consider dense traffic situations; however, various traffic conditions are evident in real-world scenarios. Therefore, we used the NGSIM I-80 and US-101 data sets to test whether the proposed TraGCAN algorithm is suitable for other traffic scenarios. The experimental results are presented in Table VI, which shows the ADE and FDE on the NGSIM I-80 and US-101 data sets. The results demonstrate that the prediction errors of all algorithms are very close, but the performance of the proposed TraGCAN algorithm still has an advantage over baselines. The NGSIM data set only contains a single-vehicle traffic agent. In addition, unlike the urban traffic environment, there is almost no heterogeneous interaction between different vehicles in the highway scenario. Note that the TraGCAN-nic algorithm does not need to consider the dynamic constraints of different traffic agents, which reduces the impact of redundant data on the final result. Therefore, the TraGCAN-nic algorithm outperformed the TraGCAN algorithm when solving the single-agent trajectory prediction problem.

[2]Administration, Next Generation Simulation, 2005. https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm.

VIII. CONCLUSION

In this article, we propose the TraGCAN algorithm to predict the trajectory of heterogeneous agents in IoV. We divide the model into two parts, i.e., the space and time dimensions, and model the motion state of traffic agents based on the GCN and LSTM. Due to the interaction between different traffic agents, we implemented a spatial attention mechanism to capture the mutual influence of different traffic agents. In addition, by considering various constraints, e.g., the type and size of different traffic agents, the proposed model can accurately predict the different behavior changes of pedestrians, bicycles, and cars. Finally, we evaluated the proposed TraGCAN algorithm on the Apolloscape autonomous driving data set. The results prove that the proposed model outperforms recent exiting methods, and the error is reduced by 15%. We have verified the versatility of TraGCAN. Based on the prediction results on the NGSIM data set, we found that the proposed

TraGCAN model demonstrated good performance for vehicle trajectory prediction on highways.

The proposed TraGCAN algorithm was designed for dense traffic situations; however, in real-world urban traffic scenarios, diverse road environments make traffic agents exhibit different motion modes. Therefore, the results in other scenarios have no obvious advantages and may not be suitable for other scenarios. Thus, in future, we plan to collect more data, combine the actual road conditions and other external information, and consider different constraints to improve the versatility of the proposed model while ensuring sufficient prediction accuracy.

REFERENCES

[1] Z. Xia, J. Wu, L. Wu, Y. Chen, J. Yang, and P. S. Yu, "A comprehensive survey of the key technologies and challenges surrounding vehicular ad hoc networks," *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 4, p. 37, 2021.

[2] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 1, pp. 90–96, Jan. 2017.

[3] C. Lin, G. Han, J. Du, T. Xu, and Y. Peng, "Adaptive traffic engineering based on active network measurement towards software defined Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3697–3706, Jun. 2021.

[4] C. Lin, G. Han, X. Qi, M. Guizani, and L. Shu, "A distributed mobile fog computing scheme for mobile delay-sensitive applications in SDN-enabled vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5481–5493, May 2020.

[5] C. Wu, Z. Liu, F. Liu, T. Yoshinaga, Y. Ji, and J. Li, "Collaborative learning of communication routes in edge-enabled multi-access vehicular environment," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 4, pp. 1155–1165, Dec. 2020.

[6] S. Moosavi, M. H. Samavatian, S. Parthasarathy, and R. Ramnath, "A countrywide traffic accident dataset," Jun. 2019, *arXiv:1906.05409*.

[7] Y. Liu, Q. Yan, and A. Alahi, "Social NCE: Contrastive learning of socially-aware motion representations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 15098–15109.

[8] F. Tang, B. Mao, N. Kato, and G. Gui, "Comprehensive survey on machine learning in vehicular network: Technology, applications and challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 2027–2057, 3rd Quart., 2021.

[9] X. Chen, C. Wu, Z. Liu, N. Zhang, and Y. Ji, "Computation offloading in beyond 5G networks: A distributed learning framework and applications," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 56–62, Apr. 2021.

[10] R. Yin, Y. Shen, H. Zhu, X. Chen, and C. Wu, "Time-critical tasks implementation in MEC based multi-robot cooperation systems," *China Commun.*, vol. 19, no. 4, pp. 199–215, Apr. 2022.

[11] X. Liu, Q. Sun, W. Lu, C. Wu, and H. Ding, "Big-data-based intelligent spectrum sensing for heterogeneous spectrum communications in 5G," *IEEE Wireless Commun.*, vol. 27, no. 5, pp. 67–73, Oct. 2020.

[12] J. A. Ansere, G. Han, and H. Wang, "A novel reliable adaptive beacon time synchronization algorithm for large-scale vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11565–11576, Dec. 2019.

[13] Y. Bi, G. Han, C. Lin, X. Wang, and Z. Liu, "Effective packet loss elimination in IP mobility support for vehicular networks," *IEEE Netw.*, vol. 34, no. 1, pp. 152–158, Jan./Feb. 2020.

[14] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2018, pp. 1–9.

[15] 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. (ITSC)*, 2014, pp. 334–341.

[16] S. Lefevre, C. Laugier, and J. Ibanezguzman, "Exploiting map information for driver intention estimation at road intersections," in *Proc. IEEE Intell. Veh. Symp.*, 2011, pp. 583–588.

[17] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2287–2301, Nov. 2011.

[18] C. Liu and M. Tomizuka, "Enabling safe freeway driving for automated vehicles," in *Proc. Amer. Control Conf. (ACC)*, 2016, pp. 3461–3467.

[19] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (GPML) toolbox," *J. Mach. Learn. Res.*, vol. 11, no. 100, pp. 3011–3015, 2010.

[20] 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, Oct. 2011.

[21] J. Schlechtriemen, F. Wirthmueller, A. Wedel, G. Breuel, and K.-D. Kuhnert, "When will it change the lane? A probabilistic regression approach for rarely occurring events," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2015, pp. 1373–1379.

[22] C. Cheng et al., "Machine-learning-aided trajectory prediction and conflict detection for Internet of aerial vehicles," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5882–5894, Apr. 2022.

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

[24] 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. (ITSC)*, 2016, pp. 2267–2272.

[25] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 961–971.

[26] C. Lin, G. Han, J. Du, T. Xu, and Z. Lv, "Spatiotemporal congestion-aware path planning toward intelligent transportation systems in software-defined smart city IoT," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8012–8024, Sep. 2020.

[27] Y. Xu, Z. Piao, and S. Gao, "Encoding crowd interaction with deep neural network for pedestrian trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5275–5284.

[28] M.-F. Chang et al., "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8740–8749.

[29] H. Xue, D. Q. Huynh, and M. Reynolds, "SS-LSTM: A hierarchical LSTM model for pedestrian trajectory prediction," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, 2018, pp. 1186–1194.

[30] G. Xie, A. Shangguan, R. Fei, W. Ji, and X. Hei, "Motion trajectory prediction based on a CNN-LSTM sequential model," *Sci. China. Inf. Sci.*, vol. 63, no. 11, 2020, Art. no. 212207.

[31] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 1–9.

[32] H. Wu, X. Li, X. Yang, and T. Wang, "Joint trajectory prediction of multi-linkage robot based on graph convolutional network," *IEEE Access*, vol. 8, pp. 221077–221092, 2020.

[33] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 1–7.

[34] J. Amirian, J.-B. Hayet, and J. Pettré, "Social ways: Learning multi-modal distributions of pedestrian trajectories with GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1–9.

[35] N. Mohajerin and M. Rohani, "Multi-step prediction of occupancy grid maps with recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10600–10608.

[36] C. Choi, "Shared cross-modal trajectory prediction for autonomous driving," 2020, *arXiv:2004.00202*.

[37] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 6120–6127.

[38] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8483–8492.

[39] H. Zhao et al., "TNT: Target-driven trajectory prediction," in *Proc. Conf. Robot Learn.*, vol. 155, Nov. 2021, pp. 895–904.

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

[41] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 261–268.

[42] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2255–2264.

Jie Li (Member, IEEE) received the B.S. degree in computer science and technology from Dalian University of Technology, Dalian, China, in 2004, and the M.S. and Ph.D. degrees in computer science and technology from Northeastern University (NEU), Shenyang, China, in 2007 and 2015, respectively.

From 2017 to 2018, she was a Visiting Scholar with the Department of Computer Science and Engineering, Pennsylvania State University, State College, PA, USA. She is currently an Associate Professor with the School of Computer Science and Engineering, NEU. Her research outcomes are published in EI/SCI indexed more than 60 publications, such as IEEE INFOCOM, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and IEEE INTERNET OF THINGS JOURNAL. She also holds a series of patents in China. Her research interests are in the area of computer networking with emphasis on design, measurements, and prototype implementation of protocols and algorithms for the Internet and IoT, mobile computing, and networking.

Han Shi is currently pursuing the M.S. degree with the Department of Computer Science and Engineering, Northeastern University, Shenyang, China.

His research interests include trajectory prediction and deep learning.

Yue Guo is currently pursuing the M.S. degree in software engineering, Northeastern University, Shenyang, China.

Her research interests include intelligent sensing and computing, and data intelligence.

Guangjie Han (Fellow, IEEE) received the Ph.D. degree from Northeastern University, Shenyang, China, in 2004.

He is currently a Professor with the Department of Internet of Things Engineering, Hohai University, Changzhou, China. In February 2008, he finished his work as a Postdoctoral Researcher with the Department of Computer Science, Chonnam National University, Gwangju, South Korea. From October 2010 to October 2011, he was a Visiting Research Scholar with Osaka University, Suita, Japan. From January 2017 to February 2017, he was a Visiting Professor with City University of Hong Kong, Hong Kong. From July 2017 to July 2020, he was a Distinguished Professor with Dalian University of Technology, Dalian, China. He has over 400 peer-reviewed journal and conference papers, in addition to 160 granted and pending patents. His current H-index is 58 and i10-index is 242 in Google Citation (Google Scholar). The total citation count of his papers raises above 12 500 times. His current research interests include Internet of Things, Industrial Internet, machine learning and artificial intelligence, mobile computing, and security and privacy.

Dr. Han has been awarded the 2020 IEEE SYSTEMS JOURNAL Annual Best Paper Award and the 2017–2019 IEEE ACCESS Outstanding Associate Editor Award. He has served on the Editorial Boards of up to ten international journals, including the IEEE SYSTEMS JOURNAL, IEEE/CAA JOURNAL OF AUTOMATICA SINICA, and *Telecommunication Systems*. He has guest-edited several special issues in IEEE journals and magazines, including the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, *IEEE Communications Magazine*, IEEE WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and *Computer Networks*. He has also served as a chair of organizing and technical committees in many international conferences. He is a Fellow of the U.K. Institution of Engineering and Technology.

Ruiyun Yu (Member, IEEE) received the bachelor's degree in mechanical engineering and the M.S. and Ph.D. degrees in computer science from Northeastern University, Shenyang, China, in 1997, 2004, and 2009, respectively.

He is currently a Professor and the Vice Dean of the Software College, Northeastern University, where he serves as the Director of Center for Cross-Media Artifificial Intelligence. He is one of the Baiqianwan Talents of Liaoning Province, China (Hundred Talents Level). His research interests include intelligent sensing and computing, computer vision, and data intelligence.

Dr. Yu is currently a member of the CCF IoT Committee and a Senior Member of CCF.

Xingwei Wang received the B.S., M.S., and Ph.D. degrees in computer science from Northeastern University, Shenyang, China, in 1989, 1992, and 1998, respectively.

He is currently a Professor with the School of Computer Science and Engineering, Northeastern University. He has authored or coauthored more than 100 journal articles, books, and book chapters, and refereed conference papers. His current research interests include future Internet, cloud computing, mobile computing, and mobile social networks.

Dr. Wang was a recipient of several best paper awards.