# Vehicle Trajectory Prediction in Connected Environments via Heterogeneous Context-Aware Graph Convolutional Networks

Yuhuan Lu, Wei Wang*, Xiping Hu*, Pengpeng Xu, Shengwei Zhou, and Ming Cai

***Abstract*—The accurate trajectory prediction of surrounding vehicles is crucial for the sustainability and safety of connected and autonomous vehicles under mixed traffic streams in the real world. The task of trajectory prediction is challenging because there are all kinds of factors affecting the motions of vehicles, such as the individual movements, the ambient driving environment especially road conditions, and the interactions with neighboring vehicles. To resolve the above issues, this work proposes a novel Heterogeneous Context-Aware Graph Convolutional Networks following the Encoder-Decoder architecture, which simultaneously extracts the hidden contexts from individual historical trajectories, varying driving scene, and inter-vehicle interactional behaviors. Specifically, the historical vehicle trajectories are fed into Temporal Convolutional Network to capture the individual context. Besides, a 2-Dimensional Convolutional Network with temporal attention is designed for transforming the scene image stream into compressing scene context. Then a Spatio-Temporal Dynamic Graph Convolutional Networks is devised to model the evolving interactional patterns, which incorporates the acquired individual and scene contexts as the representation of the node. Finally, the aforementioned three contexts are combined and fed into the decoder to produce future trajectories. The proposed model is validated on two real-world datasets which contain various driving scenarios. Results demonstrated that the proposed model outperforms state-of-the-art methods in prediction accuracy and achieves immense stability towards different vehicle states.**

***Index Terms*—Traffic big data, graph neural networks, trajectory prediction, connected vehicles, interaction context.**

## I. Introduction

THE perception of the motions of human-driving vehicles is crucial for autonomous vehicles when driving in mixed traffic streams. Such complex and ever-changing driving environments require the accurate trajectories prediction of surrounding vehicles to make decisions in advance and circumvent the potential incidents, which also enhances the mobility, sustainability, and safety of autonomous vehicles [1]. However, the exact prediction of trajectories is quite challenging especially when vehicles travel on urban road networks [2]. At this time, the motions of vehicles are affected by diverse factors, such as their own dynamics, road conditions, and interactions with surrounding vehicles. Over the past decade, researchers in the field of connected and autonomous vehicles have endeavored to address the above issues and make precise predictions of vehicle trajectories. At present, the trajectory prediction paradigms can be divided into three categories: physics-based methods, maneuver-based methods, and interaction-aware methods [3].

Physics-based methods [4] concentrate on the individual dynamic behaviors and integrate individual dynamics by linear filters to predict the future placement of vehicles. Maneuver-based methods [5] consider the limitation of driving maneuvers resulted from road conditions and incorporate the restricted maneuvers into trajectory prediction. To strengthen both the short-term and long-term features expression, physics- and maneuver-based combined methods are proposed to produce high fidelity trajectories [6]. Although the above methods achieve good performances, ignoring the impacts from surrounding vehicles still inhibits their representation ability.

With the advancement of communication technologies, vehicle-to-vehicle (V2V) enables smoother message passing between vehicles, and thus the interaction-aware methods have attracted increasingly more attention [7], [8]. Currently, methods based on Convolutional Neural Networks (CNN) are widely used to model the interactions between vehicles. Convolutional social pooling [9] constructs an Encoder-Decoder model employing the Long Short-Term Memory (LSTM) as basic cells. Within the encoder, CNN is applied to extract interdependencies from social tensors consisting of vehicle motions. Instead of predefining the size of a spatial grid, CNN-LSTM [10] designs a dynamic grid to cover the motions of vehicles and then the interaction features are seized by a two-layer CNN.

Since the great success of Graph Neural Networks (GNN) in spatio-temporal predictions [11]–[14], recent interaction-aware methods employ GNN to model the mutual effects between vehicles, which have been proved the superiority over CNN-based methods. Grip [15] proposes a graph to represent the interactions of surrounding vehicles. Then a Graph Convolutional Network (GCN) [16] is employed to capture inter-vehicle interactional patterns in spatial space. SCALE-Net [17] constructs edge characteristics for inter-vehicle measurements and then employs an edge-enhanced GCN to acquire the interaction embedding. HEAT [18] proposes an edge-enhanced Graph Attention Network (GAT) [19]

Yuhuan Lu, Wei Wang, Xiping Hu, and Ming Cai are with School of Intelligent Systems Engineering, Sun Yat-Sen University, Guangzhou 510006, China (e-mail: luyh6@mail2.sysu.edu.cn, wangw328@mail.sysu.edu.cn, huxiping@mail.sysu.edu.cn, caiming@mail.sysu.edu.cn)

Pengpeng Xu is with School of Civil Engineering and Transportation, South China University of Technology, Guangzhou 510640, China (e-mail: peng90@scut.edu.cn)

Shengwei Zhou is with State Key Laboratory of Internet of Things for Smart City and also with Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: shengwei.zhou@connect.um.edu.mo) (*Corresponding author: Wei Wang; Xiping Hu)

to handle the heterogeneity of connected vehicles and the map features extracted by gate mechanism are shared across all vehicles. While interpreting inter-vehicle interactions as a graph raises the prediction accuracy, most existing interaction-aware methods still faces the following limitations:

- Existing interaction-aware methods regarded the scene information as an invariant complement for trajectory prediction. However, the driving environment is rapidly changing and the snapshots of it at different time intervals have distinct influences on the future trajectories of vehicles.
- The most recent GNN-based methods tend to represent the connected vehicles in a static graph, which is not able to express the shape change of vehicle platoon and restrain the capability of capturing the dynamic interactional patterns.

In order to overcome the above obstructions, this work invents a new deep learning-based architecture, which explores the heterogeneous contexts inherent in the historical trajectories and ambient driving environments of the connected vehicles for emphasizing the spatio-temporal representation and prediction of vehicle mobilities. In particular, the main contributions of this work can be summarized as:

- A Heterogeneous Context-Aware Graph Convolutional Networks is proposed to consider vehicles' individual dynamics, interactional patterns, and road conditions jointly. The above three factors are mapped into context embeddings, which largely elevates the prediction accuracy.
- A dynamic scene context extraction method is designed to reinforce the expression of the effects of the driving environment on vehicle motions, where a temporal attention mechanism is incorporated into CNN to model the temporal correlations inherent in scene images.
- A novel Spatio-Temporal Dynamic Graph Convolutional Networks is developed to model the evolving inter-vehicle interactions from both the spatial and temporal domains. In the spatial domain, a node attention mechanism is designed to insert the scene context into node feature. And in the temporal domain, a Evolving Graph Convolutional Network module is developed to capture the temporal dynamics of the connected vehicle platoon.

The remainder of this paper is organized as follows: Sec. II reviews the related works. Sec. III provides an overview of this work. Sec. IV elaborates the proposed Heterogeneous Context-Aware Graph Convolutional Networks. Sec. V validates the proposed model on two real-world driving dataset. Finally, Sec. VI concludes this work and points out some future research directions.

## II. Related Work

Accounting for the deep-seated interactional behaviors is most important for the high-accuracy trajectory prediction and scholars have applied various machine learning-based methods to raise the interpretation. This section briefly introduces some recent works on trajectory prediction with two types of prevalent interaction representations.

### A. Euclidean Representation

Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are two most widely used approaches to reflect the interactional behaviors with regard to Euclidean representation. LSTM-based trajectory prediction [20] feeds the historical coordinates of all vehicles into the corresponding single LSTM cell and obtain the motion context for each grid. Next, a unified LSTM network produces the occupancy probability for each grid over these contexts and predict the future trajectories of vehicles. GRU-based trajectory prediction [21] employs the lightweight variant of LSTM to learn from the historical trajectory data. As Gated Recurring Unit (GRU) has less number of parameters than LSTM, the training of it is more simple and fluent, and thus it is suitable for short-term trajectory prediction. ARNN [22] is the improved model upon the vanilla RNN, which incorporates the self-attention mechanism to further capture the long-term dependencies inherent in the trajectory data. Through the attention mechanism, the interaction between historical trajectory information and current traffic state is exploited to enhance the forecasting performance. S2TNet [23] makes use of the multi-head attention mechanism to merge the spatial and temporal features of historical trajectories and achieves plausible results. Convolutional social pooling [9] devises a pooling grid over the vehicles in the autonomous scene. Based on the Long Short-Term Memory (LSTM) cells, vehicle motions are captured by the stacked LSTM encoder and then a social tensor cultivated by CNN and pooling layers is employed to model the inter-dependencies of all vehicles. Finally, a maneuver activated LSTM decoder generates the future motions over a given time window. CNN-LSTM [10] also utilizes the LSTM encoder to seize the sequential features of vehicle motions. Based on a designed dynamic 3×3 grid over the vehicular group, a CNN and LSTM combined interaction extractor is proposed to fully represent the inter-vehicle influences. MATF [24] first introduces the tensor realization into the spatial constraints modeling of surrounding vehicles. A multi-faceted tensor is constructed which takes the scene variances and dynamic motions into account. To learn the spatial inter-dependencies, the built tensor is fed into a U-Net-like architecture while retaining the spatial resolution of trajectories.

The above-mentioned trajectory prediction methods are able to capture dynamic interactional behaviors of vehicles to some extent but fail to comprehensively cover the heterogeneity of relationships among vehicles due to the narrow geometric-structure representation ability of Euclidean machine learning.

### B. Non-Euclidean Representation

Data represented in the form of graph is ubiquitous in the smart city, and increasing more recent works focus on the development of Graph Neural Networks (GNN) to address the problem of representation learning upon non-euclidean data. GNNs are categorized into either spectral method [16], [25], [26] or non-spectral method [19], [27], [28]. Spectral GNN, e.g., Graph Convolutional Network (GCN) [16] generalize the convolutional operation from grid data to graph data, which applies the graph Laplacian matrix to aggregating the information of nodes. On the other hand, non-spectral GNN, e.g., Graph Attention Network (GAT) [19] restricts the information aggregation to performing on local nodes, which reduces the computational overhead on Laplacian eigenbasis. Motivated by the excellent performance of GNNs, some recent works attempt to utilize GNNs to explore the applications of smart city, especially the spatio-temporal modeling. ST-GDN [29] is a hierarchically structured GCN, which learns from both the local geographical dependencies and the global spatial semantics to improve the citywide traffic flow prediction. ADGCN [30] empowers the combination of attention mechanism and GCN to tackle the congestion recognition problem. To catch the long-range evolution of congestion, GCN is applied on a digraph to model the high-order spatio-temporal features. ST-RGAN [31] develops a spatio-temporal block by the multi-faced GAT to capture both spatial and temporal dependencies of the turn-level road network topology simultaneously. These GNNs can handle heterogeneous dependencies in a graph and achieve great performance on spatio-temporal modeling.

Graph is exactly the most suitable structure for representing the interactions among vehicles, and therefore the above GNN techniques advance the expression of complex inter-dependencies and relationships among objects [32]. GRIP [15] constructs an undirected graph containing the nodes and edges referring to the vehicles and interactions, respectively. The node feature is constituted by a sequence of coordinates over the observed time steps. To extract the interaction contexts, a Graph Convolutional Network (GCN) is employed with spatially correlated graph operations. DGG [33] uses a two-stream GNNs combined with LSTM networks to capture the spatio-temporal motions patterns. Based on the spectral regularization, the fusion of patterns is transformed into the future trajectories. SCALE-Net [17] regards the edge characteristics of a connected graph as the inter-vehicle measurements. An edge-enhanced attention mechanism is invented to assign the importance weights to different vehicles and then apply the GCN to integrate the features and transfer the features to predictive motions. VectorNet [34] proposes a novel hierarchical framework to amplify the interaction modeling. The first layer is composed of multiple sub-graphs, each of which corresponds to a specific vehicle and the second layer combines all sub-graphs and represent the inter-dependencies by a fully-connected graph.

GNN-based interaction modeling has superior capability of heterogeneity representation over the Euclidean modeling. However, the existing works lack the combination of global and local contexts, which results in the static-excessive representation.

## III. Overview

### A. Problem Formulation

The overarching goal of this work is to learn the underlying contexts regarding interactional behaviors of connected vehicles under various traffic situations and to forecast the future trajectory of the target vehicle accordingly.

Suppose the historical states of the networked vehicle $i$ at time interval $t$ is $\mathcal{HS}_t^i = \left\{hs_{t-T+1}^i, hs_{t-T+2}^i, \dots, hs_t^i\right\}$, where $T$ denotes the traceback time window. Considering the trade-off between communication efficiency and shared-information richness, an instantaneous state $hs_t^i$ contains both the position and velocity of the vehicle $i$ at time interval $t$. Accordingly, the historical states of all networked vehicles is represented by $\mathcal{HS}_t = \bigcup_{i=0}^{N_t-1} \mathcal{HS}_t^i$, with $i = 0$ as the target vehicle and $N_t$ as the number of connected vehicles at time interval $t$. While involving the scene context $\mathcal{SC}_t$ derived from the local maps at $t$-th time interval, the input to the model can be defined as follows:

$$\mathbf{HD}_t = \{\mathcal{HS}_t, \mathcal{SC}_t\} \tag{1}$$

From this, the output from the model is the predicted trajectory of target vehicle as follows:

$$\mathbf{PT}_t = \left\{\left(x_{t+1}^0, y_{t+1}^0\right), \left(x_{t+2}^0, y_{t+2}^0\right), \dots, \left(x_{t+\tau}^0, y_{t+\tau}^0\right)\right\} \tag{2}$$

where $\left(x_{t+1}^0, y_{t+1}^0\right)$ refers to the 2-Dimensional coordinates of target vehicle and $\tau$ stands for the predictive time window.

### B. The Proposed Model

In order to accurately predict the future trajectory of a target vehicle, a Heterogeneous Context-Aware Graph Convolutional Network is proposed to extract the hidden contexts of the target and surrounding vehicles from their historical vehicle trajectories along with the ambient driving environment. Fig. 1 presents the framework of the proposed model which follows an Encoder-Decoder architecture. The raw input data containing historical vehicle trajectories and road features are firstly transmitted to two-stream neural networks. On the one hand, a Temporal Convolutional Network replacing the traditional RNN is employed to capture the sequential features for each vehicle and transform the features into individual context embedding. On the other hand, a 2-Dimensional Convolutional Network is devised to map the driving scenarios images into scene context. At the same time, a connected graph is constructed with the hyperbolic structure where the central node denotes the target vehicle and outside nodes refer to the surrounding vehicles. Afterward, a Spatio-Temporal Dynamic Graph Convolutional Network (STDGCN) is developed to model both the spatial and temporal evolutions of interactions over the above connected graph. The proposed STDGCN incorporates both the individual and scene contexts as the representation of the node, which largely enhances the mining of interaction context. Finally, the derived three contexts are fused and fed into a Temporal Convolutional Network decoder to forecast the future trajectory of the target vehicle.

## IV. Heterogeneous Context-Aware Graph Convolutional Networks

In this section, we elaborate on the capture of three distinct but closely linked contexts (i.e., individual context, scene context, and interaction context) which are essential for the proposed trajectory prediction framework.

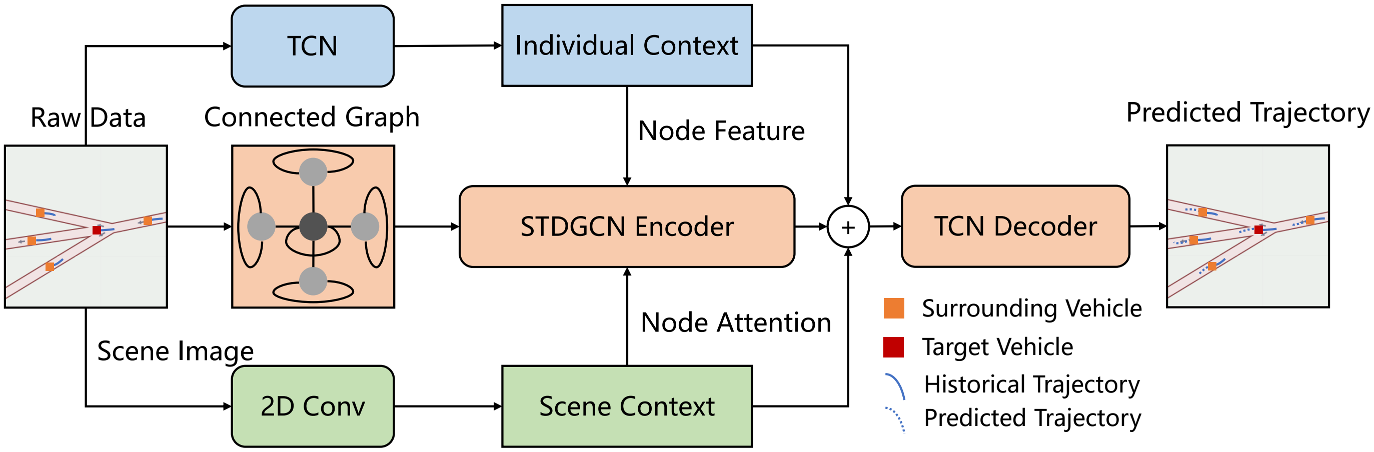


Fig. 1. The framework of HCAGCN. **TCN**: Temporal Convolutional Network. **2D Conv**: 2-Dimensional Convolution. **STDGCN**: Spatio-Temporal Dynamic Graph Convolutional Network.

### A. Individual Context Extraction

Individual context is a deep-seated understanding of the behavioral patterns of vehicles and thus it is crucial to the forecast of future vehicle positions. Nevertheless, context information is intensely hidden in a historical trajectory and therefore a suitable time-series processor is needed to seize the sequential context.

Recurrent Neural Networks (RNNs) are specialized for handling sequential dependency which makes them the default choices to resolve the time-series problems [35], [36]. However, inspired by the recent studies, some novel CNN-based models are more effective than RNN-based approaches across a variety of tasks, such as machine translation [37], signal processing [38] and traffic prediction [39]. Encouraged by the success of the Convolutional Neural Network (CNN) family in sequential modeling, a Temporal Convolutional Network (TCN) model with dilated causal convolution is applied to learning the temporal dependency inherent in individual trajectory and then converting it into individual context. Compared with RNNs, TCN is more suitable for the task of context extraction, because: 1) The activations within each layer of TCN are independent. Such a design greatly improves the training speed, which has direct impact on the efficiency of context embedding. 2) Through adjusting the parameters of dilated filters, the TCN output can be restricted to relying more on short-range input time steps, the property of which exactly satisfies the demand of trajectory prediction framework [40].

As shown in Fig. 2a, TCN exerts the dilated causal convolution to enlarge the receptive field of network without extending the filter size. Particularly, the dilated factor $d$ increases exponentially with layer depth. Let $f_l : \{0, 1, \dots, k-1\}$ be the filter set of the $l$-th layer. The output activation of the $l$-th layer at time interval $t$ is given by:

$$F_l(t) = \sum_{q=0}^{k-1} f_l(q) F_{l-1}(t - q * d) \tag{3}$$

where $f_l(q)$ denotes the $q$-th filter in the corresponding filter set, $k$ refers to the filter size and $F_{l-1}(t - q * d)$ is the input activation from the last layer $(l-1)$. To further capture the individual context, the dilated convolution layers are stacked in a blocky structure as presented in Fig. 2b. Especially, the dilated TCN utilizes the skip connection [41] to mitigate the gradient vanishing and the network degradation. Suppose the output activation at time interval $t$ of $j$-th block is denoted by $F^j(t) \in \mathbb{R}^{F_I}$, where $F_I$ denotes the embedding size of individual context. The final output is obtained as follows:

$$C(t) = ReLU\left(\sum_{j=1}^{B} F^j(t)\right) \tag{4}$$

where $B$ is the number of blocks in TCN and $ReLU(\cdot)$ is the Rectified Linear Unit [42] applied in the activation layer:

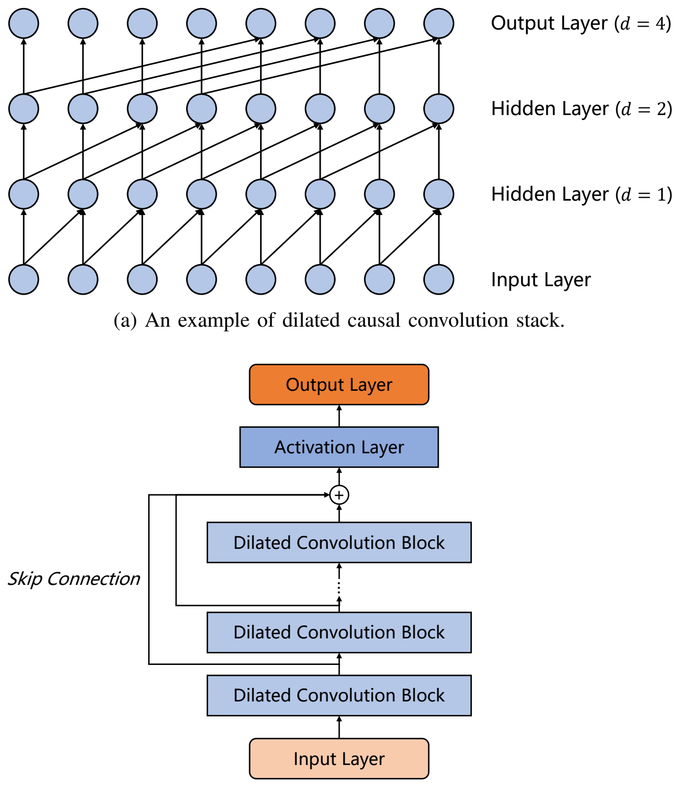
$$ReLU(x) = \max(0, x) \tag{5}$$

For convenient, the derived context embedding of a specific vehicle $i$ at time interval $t$ is denoted by $Z_i^I(t)$. Here, $Z_i^I(t) = C(t) \in \mathbb{R}^{F_I}$.

### B. Scene Context Extraction

The driving environment is a vital factor highly affecting the vehicle trajectory and therefore the scene context should be appropriately extracted from the road condition to represent the latent information of the driving environment. Recently, many studies have explored all sorts of representation methods of scene context. ReCoG [43] employs a CNN to encode the local map into context embedding. However, such a local map is centered at the target vehicle's initial position and overlooks the real-time motions. MATF [24] constructs a multi-agent tensor that retains the encoded scene image in the channel dimension and then applies CNN to capture the fusion of multi-agent interactions and scene context. Notably, in MATF, the scene image snapshots at different time intervals have the same impact on future trajectories of target vehicles.

To exploit a more informative context embedding, an attention mechanism is designed to model the dynamics of temporal correlations between different time intervals in the scene



(a) An example of dilated causal convolution stack.

(b) The dilated TCN model employs blocky-stacked dilated convolutions.

Fig. 2. Illustration of dilated causal convolution and TCN model.

images. Referring to the channel attention [44] in CNN for emphasizing the feature representation, a temporal attention is developed to adaptively select the scene images at the most relevant time intervals to produce the scene context. Assume that the scene-image series of vehicle $i$ at current time interval $t$ is denoted by a three-order tensor $\mathcal{M}_i^t \in \mathbb{R}^{T_S \times H \times W}$, where $T_S$ denotes the input time window of scene images, $H$ and $W$ denotes the height and width of an image, respectively. Tensor $\mathcal{M}_i^t$ is first reshaped to $\tilde{\mathcal{M}}_i^t \in \mathbb{R}^{T_S \times P}$, and then a matrix multiplication is applied between $\tilde{\mathcal{M}}_i^t$ and its transpose. At last, a softmax function is performed to derive the temporal attention matrix $\mathbf{A} \in \mathbb{R}^{T_S \times T_S}$:

$$\alpha_{kj} = \frac{\exp\left(\left(\tilde{\mathcal{M}}_i^t\right)_j \cdot \left(\tilde{\mathcal{M}}_i^t\right)_k\right)}{\sum_{j=1}^{T_S} \exp\left(\left(\tilde{\mathcal{M}}_i^t\right)_j \cdot \left(\tilde{\mathcal{M}}_i^t\right)_k\right)} \tag{6}$$

where $\left(\tilde{\mathcal{M}}_i^t\right)_j \in \mathbb{R}^P$ is the transpose of $j$-th row in $\tilde{\mathcal{M}}_i^t$. $\alpha_{kj}$ semantically represents the impact from the $j$-th time interval on the $k$-th time interval. Furthermore, a matrix multiplication is likewise employed between the $\mathbf{A}$'s transpose and $\tilde{\mathcal{M}}_i^t$. The result is then reshaped to $\mathbb{R}^{T_S \times H \times W}$. Finally, the reshaped result is scaled by a parameter $\beta$ and a skip connection is also executed to obtain the contextual output $\vec{\mathcal{M}}_i^t \in \mathbb{R}^{T_S \times H \times W}$:

$$\left(\left(\vec{\mathcal{M}}_i^t\right)_{[1]}\right)_k = \beta \sum_{j=1}^{T_S} \left(\alpha_{kj} \left(\tilde{\mathcal{M}}_i^t\right)_j\right) + \left(\tilde{\mathcal{M}}_i^t\right)_k \tag{7}$$

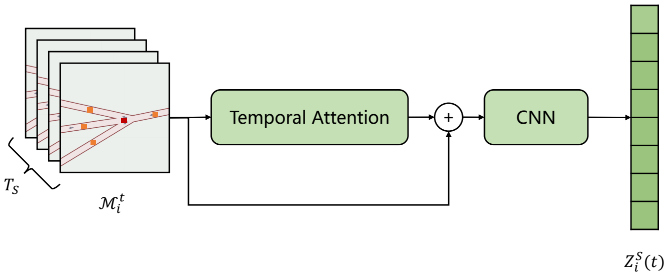


Fig. 3. The workflow of scene context extraction.

where $\beta$ is a learnable parameter and $\left(\vec{\mathcal{M}}_i^t\right)_{[1]}$ is the 1-mode unfolding of $\vec{\mathcal{M}}_i^t$. Through the above temporal attention, each time interval (namely channel) of final output $\vec{\mathcal{M}}_i^t$ is a weighted sum of all time intervals, which enhances the expression of dynamical scene features. To achieve the context embedding, a CNN is applied to the refined scene tensor:

$$Z_i^S(t) = \mathrm{FC}(\mathrm{CNN}(\vec{\mathcal{M}}_i^t)) \tag{8}$$

Here, $\mathrm{FC}(\cdot)$ stands for the fully connected layer to obtain the weighted summation of features produced by CNN. The complete workflow with regards to context extraction is presented in Fig. 3.

### *C. Interaction Context Extraction*

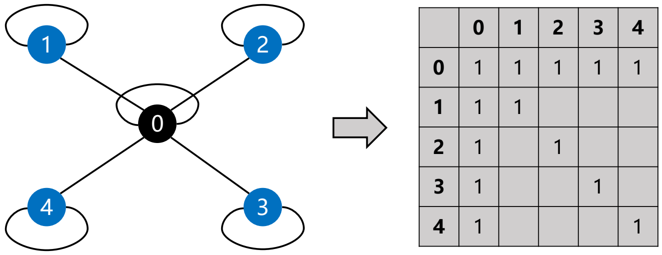
The graph is naturally an excellent data structure to represent the interactions among connected vehicles [18]. Benefiting from the advancement of Graph Convolutional Network (GCN), recent studies have employed a wide variety of GCN-based approaches to recognize the interactional patterns [15], [34], [45]. Most of them achieve prominent performances over trajectory prediction.

Without loss of generality, the connected graph at time interval $t$ is denoted as $\mathcal{G}_t\left(\mathcal{V}_t, \mathcal{E}_t\right)$. $\mathcal{V}_t$ is the set of nodes at time interval $t$ and each node corresponds to a real-world vehicle. $\mathcal{E}_t \subseteq |\mathcal{V}_t| \times |\mathcal{V}_t|$ denotes the set of edges at the same time. The portrait of connected graph construction is exhibited in Fig. 4. Let $H_t^l \in \mathbb{R}^{N_t \times F_I}$ be the activation in the $l$-th layer at time interval $t$, where $N_t$ is the number of nodes at time interval $t$ and $F_I$ is the size of input individual context as described in Section IV-A. According to the classical GCN model proposed by Kipf and Welling [16], the propagation rule between two consecutive graph convolution layers is as follows:

$$H_t^{l+1} = f\left(D^{-\frac{1}{2}} E D^{-\frac{1}{2}} H_t^l W_t^l\right) \tag{9}$$

where $E \in \mathbb{R}^{N_t \times N_t}$ denotes the adjacency matrix of connected graph $\mathcal{G}$ with self-loop. $D \in \mathbb{R}^{N_t \times N_t}$ is the degree matrix of $E$ wherein $D_{jj} = \sum_k E_{jk}$. $W_t^l \in \mathbb{R}^{F_I \times F_I}$ is a layer-specific weighted matrix and $f(\cdot)$ refers to an activation function. Additionally, $H_t^0 = \left[Z_0^I(t), Z_1^I(t), \ldots, Z_{N_t-1}^I(t)\right]^{\mathrm{T}}$. $Z_i^I(t)$ is computed by Eq. (4) and $Z_0^I(t)$ represents the individual context of target vehicle.

Although the classical GCN is capable of modeling the static inter-vehicle correlations, a connected vehicle platoon



|   | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 |   |   |   |
| 2 | 1 |   | 1 |   |   |
| 3 | 1 |   |   | 1 |   |
| 4 | 1 |   |   |   | 1 |

Fig. 4. Illustration of connected graph construction and its corresponding adjacency matrix. The black and blue circles represent the target and surrounding vehicles respectively.

often dynamically evolves. For example, the number of vehicles in a connected vehicle platoon changes over time due to the entry and exit of vehicles; hence, the connected graph should be updated accordingly to reflect the temporal variation of interactional patterns. Similarly, the driving environment of a platoon is constantly shifting since the high-speed movement of vehicles. Thus the dynamics of such spatial information should be captured to augment the interaction context extraction.

To resolve the above issues, a dynamic mechanism is developed that is imposed on GCN to capture the evolution of interactional patterns. Specifically, a Spatio-Temporal Dynamic Graph Convolutional Network is devised to capture the dynamics of the connected graph from both the spatial and temporal domains.

In the spatial domain, a node attention mechanism is designed to adapt the varying scene context into the connected graph, which strengthens the representation ability of vanilla GCN. Given the scene context $Z_i^S(t)$ of vehicle $i$ at time interval $t$, the node attention is calculated as follows. Likewise, $Z_0^S(t)$ refers to the scene context of the target vehicle.

$$r_t^i = v^{\mathrm{T}} \tanh\left(W_r Z_i^I(t) + U_r Z_i^S(t) + b_r\right) \tag{10}$$

$$\gamma_t^i = \frac{\exp\left(r_t^i\right)}{\sum_{k=0}^{N_t-1} \exp\left(r_t^k\right)} \tag{11}$$

where $v$, $b_r$, $W_r$ and $U_r$ are learnable parameters. Eq. (11) is a softmax function ensuring that all the attention weights obtained in Eq. (10) sum to one. In this case, attention weight semantically represents the effect intensity of scene context on vehicle stage. After deriving the attention scores, the node feature of vehicle $i$ at time interval $t$ can be updated as follows:

$$\tilde{Z}_i^I(t) = \gamma_t^i Z_i^I(t) \tag{12}$$

In the temporal domain, the principal difficulty is how to inject the temporal dynamics into the parameters of conventional GCN and then constitute an evolving graph sequence. Fortunately, recurrent architecture naturally fulfills the above requirements [46]. In terms of the selection of recurrent models, this work utilizes the Gated Recurrent Unit (GRU) as an alternative to Long Short-Term Memory (LSTM) network [47]. Contrasted with LSTM, GRU is more computationally efficient [48] and has been widely used in modeling the time-varying characteristics of trajectory [40], [49], [50]. To dynamically update the weighted matrix $W_t^l$ of GCN based on current and historical contexts, $W_t^l$ is regarded as the hidden state of GRU. Consequently, the activations and hidden states in GRU should be extended from vectors to matrices:

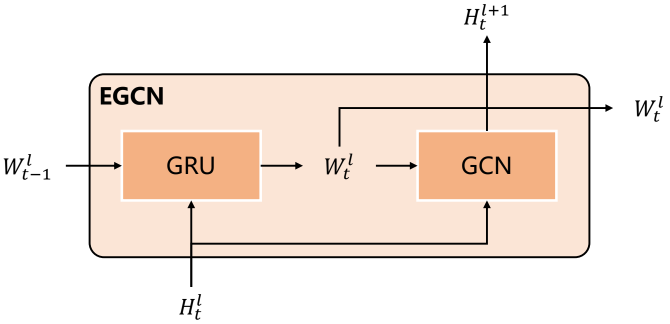


Fig. 5. Illustration of Evolving Graph Convolutional Network (EGCN) module for capturing temporal dynamics of connected graph.

$$\begin{aligned} P_t &= \mathrm{sigmoid}\left(W_P H_t^l + U_P W_{t-1}^l + B_P\right) \\ Q_t &= \mathrm{sigmoid}\left(W_Q H_t^l + U_Q W_{t-1}^l + B_Q\right) \\ \tilde{W}_t^l &= \tanh\left(W_0 H_t^l + U_0\left(Q_t \circ W_{t-1}^l\right) + B_0\right) \\ W_t^l &= \left(1 - P_t\right) \circ W_{t-1}^l + P_t \circ \tilde{W}_t^l \end{aligned} \tag{13}$$

where $W_P$, $W_Q$, $W_0$, $U_P$, $U_Q$, and $U_0$ are all learnable parameters. $B_P$, $B_Q$, and $B_0$ are trainable bias terms. $\circ$ refers to the Hadamard product. Notably, in Eq. 13, the number of columns of activation $H_t^l$ is required to be the same as that of weighted matrix $W_{t-1}^l$. Since the initial node feature in the connected graph is produced by the combination of individual context and node attention, the size of the feature is able to retain at $F_I$, which is identical to the number of columns of $W_{t-1}^l$. The structure of Evolving Graph Convolutional Network (EGCN) module is presented in Fig. 5.

On the basis of the aforementioned node attention and EGCN module, a Spatio-Temporal Dynamic Graph Convolutional Network (STDGCN) is constructed which integrates the spatial and temporal domain to extract the profound interaction context. The architecture of STDGCN is illustrated in Fig. 6. Specifically, STDGCN consists of a node attention block and two EGCN modules. The historical individual contexts $\left\{Z^I(t-T+1), Z^I(t-T+2), \ldots, Z^I(t)\right\}$ are first fed into the node attention block, where $Z^I(t) = \left[Z_0^I(t), Z_1^I(t), \ldots, Z_{N_t-1}^I(t)\right]^{\mathrm{T}}$. Then the filtered contexts $\left\{\tilde{Z}^I(t-T+1), \tilde{Z}^I(t-T+2), \ldots, \tilde{Z}^I(t)\right\}$ are sequentially fed into the EGCNs to seize the spatio-temporal ensemble dynamics. The outputs of lower EGCN are inputted to the upper EGCN to figure out high-level features. Finally, a fully connected layer is utilized to integrate features and transform the integration into an interaction context.

### *D. Trajectory Prediction & Model Training*

In the encoder, embeddings from the branches of individual context, scene context, and interaction context are aggregated as the new input to the decoder:

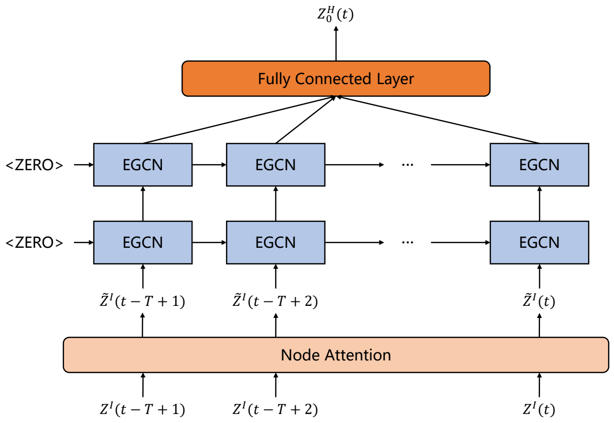


Fig. 6. Overview of Spatio-Temporal Dynamic Graph Convolutional Network (STDGCN) for the interaction context extraction.

$$Z_0(t)=\left[Z_0^I(t);Z_0^S(t);Z_0^H(t)\right] \tag{14}$$

where $Z_0^H(t)\in\mathbb{R}^{F_H}$ denotes the interaction context of target vehicle extracted by STDGCN (as shown in Fig. 6) and $[\cdot;\cdot]$ refers to the concatenation operation.

In the decoder, a TCN model similar to Section IV-A is applied to make final predictions:

$$\mathbf{PT}_t=\mathrm{TCN}_{\mathrm{dec}}\left(Z_0(t)\right) \tag{15}$$

As the proposed HCAGCN is smooth and differentiable, back-propagation [51] is able to be employed in this work to train the approach, as shown in Algorithm 1. During the training stage, parameters in HCAGCN are optimized by minimizing Huber Loss [52], which combines the superb properties from both Mean Absolute Error (MAE) and Mean Square Error (MSE). Specifically, $N$ training samples are selected, each of which contains $\tau$ prediction steps, to optimize the model. Therefore, the loss function is defined as:

$$\begin{aligned}Loss=&\frac{1}{N\tau}\sum_{j=1}^{N}\sum_{k=1}^{\tau}\mathbb{I}_{\left\|pt_{t_j+k}-\hat{pt}_{t_j+k}\right\|_2\leq\delta}\frac{\left\|pt_{t_j+k}-\hat{pt}_{t_j+k}\right\|_2^2}{2}\\&+\mathbb{I}_{\left\|pt_{t_j+k}-\hat{pt}_{t_j+k}\right\|_2>\delta}\left(\delta\left\|pt_{t_j+k}-\hat{pt}_{t_j+k}\right\|_2-\frac{1}{2}\delta^2\right)\end{aligned} \tag{16}$$

where $pt_t$ is the predicted position (namely 2-Dimensional coordinates) of target vehicle at time interval $t$ and $\hat{pt}_t$ is the ground-truth position of it. $\delta$ is set to 1 and $\|\cdot\|_2$ denotes the L2-norm.

## V. Experimental Results

This section examines the proposed HCAGCN model on two real-world datasets. Firstly, experimental settings are introduced and parametric studies are executed to find the optimal parameter combination. Then the proposed model is compared with representative approaches in various scenarios. Eventually, ablation studies are done to verify the effectiveness of different components in HCAGCN.

**Algorithm 1:** Training of HCAGCN

```
Input: The set of vehicle trajectories HD
Output: Model parameters W
$Z^I=\varnothing$
$Z^S=\varnothing$
$Z^H=\varnothing$
for $i\in epoch$ do
    $Z_i=\varnothing$
    $Z_i^I=\mathrm{TCN}(\mathbf{HD}_i)$
    $Z_i^S=\mathrm{FC}(\mathrm{CNN}(\vec{\mathcal{M}}_i)))$
    $Z_i^H=\mathrm{STDGCN}\left(Z_i^I,Z_i^S\right)$
    for $t\in\mathbf{T}$ do
        $Z_i(t)=\left[Z_i^I(t);Z_i^S(t);Z_i^H(t)\right]$
        $Z_i=\mathrm{concat}\left(Z_i,Z_i(t)\right)$
    end
    $\mathbf{PT}_i=\mathrm{TCN}_{dec}(Z_i)$
    find the error $e=|\mathbf{PT}_i-\mathbf{PT}_{real}|$
    if $|e|>e_{min}$ then
        Update all weights in W
    else
        return W
    end
end
```

### A. Experimental Settings

**Datasets**: To evaluate the proposed model, two prevalent public accessible datasets are used.

*1) INTERACTION Dataset* [53]: This dataset provides naturalistic driving states of various traffic participants from different countries. At a given timestamp, the state of a vehicle contains its position and velocity. To fully investigate the HCAGCN, three kinds of highly interactive driving scenarios are considered in this work, namely roundabout, highway ramp and un-signalized intersection. Since there are only a few instances corresponding to each of the above scenario types in raw dataset, for increasing the richness of driving scene, vehicle trajectories in Xuancheng city, China, collected by *VSensor*[1] are accommodated into INTERACTION dataset. Ultimately, the dataset is split into the training set, validation set, and testing set suggested by the authors of INTERACTION, which contains 311,275 pieces, 103,758 pieces, and 102,961 pieces, respectively.

*2) NGSIM US-101 Dataset* [54]: This dataset is collected under the Next Generation SIMulation (NGSIM) program, which records the motions of vehicles from an arterial road segment of the U.S. Highway 101. Since this dataset is dominated by the scenario of lane-keeping, this work reconstructs a refined dataset with 15,412 pieces for lane-keeping and lane-changing scenarios, respectively. After the above procedure, a total of 30,824 pieces is randomly split into a training set (24,659 pieces) and a validation set (6,165 pieces). Referring to the recent works [18], [43] that points out the shortage of traffic scenarios in this dataset, NGSIM US-101 is only

[1] https://vsensor.openits.cn/

utilized as the validation set for parametric selection in the experiments.

**Evaluation Metrics**: Following prior works [20], [55], the prediction performance is evaluated by two widely adopted error metrics in meter:

*1) Average Displacement Error (ADE)*: This metric exercises the average Euclidean distance between the predicted trajectories and the ground-truth trajectories of all vehicles:

$$\text{ADE} = \frac{\sum_{i=1}^{M} \sum_{t=t_0+1}^{t_0+\psi} \left\| pt_t^i - \hat{pt}_t^i \right\|_2}{M\psi} \tag{17}$$

*2) Final Displacement Error (FDE)*: This metric is calculated by the average Euclidean distance between the predicted trajectories and the ground-truth trajectories of all vehicles at final positions:

$$\text{FDE} = \frac{\sum_{i=1}^{M} \left\| pt_{t_0+\psi}^i - \hat{pt}_{t_0+\psi}^i \right\|_2}{M} \tag{18}$$

**Implementation Details**: The proposed model is executed by using the PyTorch [56] and the GNN in this model is realized through PyTorch Geometric [57]. In the training stage, Adam optimizer with Cyclical Learning Rates (CLR) [58] is used to minimize the training loss and the optimized learning rate is $1.52e^{-03}$. Simultaneously, the batch size is set to 128. To prevent over-fitting, the stop early strategy is employed to automatically determine the number of epochs, which stops the training procedure when the training loss decreases in 10 consecutive epochs while the validation loss increases at the same time. In addition, the traceback time window $T$ and the predictive time window $\tau$ is set to 32 (3.2 seconds) and 16 (1.6 seconds) time steps, respectively. The embedding size $F_I$ is set as 16. A vehicle is selected as the surrounding vehicle of a given target vehicle when its distance to the target vehicle is within 30 meters.

All experiments are conducted on a machine equipped with a GeForce RTX 3090 GPU and the physical memory of the GPU is 24 GB.

## *B. Parametric Studies*

Network structure definitely has an impact on forecast performance. Therefore, this section investigates the effects of four structure-related hyperparameters on the trajectory prediction to identify the optimal hyperparameters combination. The prediction results on the two datasets with different parametric values are shown in Table I.

TABLE I
PREDICTION RESULTS OF THE PROPOSED HCAGCN WITH RESPECT TO STRUCTURE-RELATED HYPERPARAMETERS.

| | INTERACTION | | NGSIM US-101 | |
|---|---|---|---|---|
| | ADE (m) | FDE (m) | ADE (m) | FDE (m) |
| $d = 2$ | 0.22 | 0.69 | 0.18 | 0.60 |
| $d = 4$ | 0.18 | 0.57 | 0.12 | 0.38 |
| $d = 8$ | 0.17 | 0.57 | 0.12 | 0.37 |
| $L_I = 1$ | 0.24 | 0.72 | 0.17 | 0.55 |
| $L_I = 3$ | 0.18 | 0.57 | 0.12 | 0.38 |
| $L_I = 5$ | 0.18 | 0.56 | 0.11 | 0.36 |
| $L_S = 1$ | 0.20 | 0.61 | 0.13 | 0.42 |
| $L_S = 2$ | 0.18 | 0.57 | 0.12 | 0.38 |
| $L_S = 3$ | 0.21 | 0.64 | 0.15 | 0.49 |
| $L_H = 1$ | 0.27 | 0.85 | 0.22 | 0.71 |
| $L_H = 2$ | 0.18 | 0.57 | 0.12 | 0.38 |
| $L_H = 3$ | 0.20 | 0.63 | 0.13 | 0.40 |

**Dilated Factor *d***: Dilated factor determines the receptive field of the model and then affects the representation of individual context. In this experiment, 2, 4, and 8 are selected as the candidate values for $d$. The results show that $d = 4$ or 8 is a much better choice than $d = 2$ for both INTERACTION and NGSIM US-101 datasets. However, the increase of dilated factors also expands the computational burden. Thus $d = 4$ is a trade-off choice between computational efficiency and prediction performance.

**The Number of Dilated Convolution Blocks $L_I$**: Analogous to the results of dilated factor $d$, HCAGCN achieves better performance when $L_I = 3$ or 5 than $L_I = 1$. This implies that the stack of dilated convolution blocks indeed captures the deep-seated and meaningful individual context embedding. It can be observed that the gaps between the results produced by $L_I = 3$ and $L_I = 5$ are small. Accordingly, $L_I = 3$ is selected as the efficiency-accuracy balance.

**Number of Convolution Layers $L_S$**: Likewise, the stack of convolution layers is able to extract high-level embedding of scene context. It can be found that $L_S = 2$ has the best results on both INTERACTION dataset and NGSIM US-101 dataset. Notably, the performance for $L_S = 3$ is even worse than that for $L_S = 1$, which indicates that the excessive number of layers of CNN may cause over-fitting and then result in the degradation of generalization.

**Number of EGCN Layers $L_H$**: EGCN is utilized in modeling the spatio-temporal dynamics of interactional behaviors. When the number of layers $L_H = 2$, the proposed model produces the best results. This indicates that STDGCN may be under-fitting or over-fitting for $L_H = 1$ or $L_H = 3$, respectively. Therefore, $L_H = 2$ is the most suitable setting for STDGCN.

To sum up, the optimal hyperparameters combination is $d = 4$, $L_I = 3$, $L_S = 2$ and $L_H = 2$, which is applied in the following experiments.

## *C. Model Comparison*

In this section, the proposed HCAGCN is compared with six well-known baseline models on the INTERACTION dataset.

**ARIMA** [59]: Autoregressive Integrated Moving Average model is a statistical model that predicts the future points in the time-series data.

**CS-LSTM** [9]: This model utilizes an LSTM encoder-decoder with convolutional social pooling to improve the robust learning of interdependencies in processing vehicle trajectories.

**CNN-LSTM** [10]: This model employs CNN and LSTM to model the interactions and dynamics of vehicles, respectively.

**VectorNet** [34]: VectorNet constructs a hierarchical GNN that first represents the individual road components as vectors and then exploits the high-order interactions among all components.

**Social-WaGDAT** [60]: This model designs a Graph Double-Attention Network that can capture the spatio-temporal dynamics of connected vehicles and model the complex interactions among vehicles by message passing.

**EvolveGraph** [61]: EvolveGraph is designed for multi-agent trajectories prediction, which evolves the latent interaction graphs to represent the uncertainty of future behaviors.

Table. II shows the prediction results of the above baselines and the proposed HCAGCN. To investigate the model performances in-depth, approaches are evaluated in various scenarios. Notably, the best results (lowest ADE and FDE) are highlighted in bold.

It can be observed clearly that deep learning models achieve much better accuracy than classical machine learning models (ARIMA). This suggests that deep learning approaches have advantages in modeling heterogeneous trajectory data. Conventional deep learning models (CS-LSTM and CNN-LSTM) that utilize CNN and LSTM to exploit interactions are capable of offering decent prediction results. Nevertheless, the GNN-based model (VectorNet) accomplishes higher accuracy in all scenarios. It indicates that GNN is superior to conventional deep learning models in terms of recognizing the interactional patterns. In the highway ramp scenario, three dynamic GNN-based models (Social-WaGDAT, EvolveGraph, and HCAGCN) tremendously outperform VectorNet. The main reason is that the shape of the vehicle platoon often changes dramatically when passing the highway ramp and static GNN fails to represent such transient dynamics of the connected graph. Additionally, Social-WaGDAT and HCAGCN both enhance the individual context integration into the GNN interpreter and thus they have better performances in un-signalized intersections where the speed of traffic flow is smoother and steadier than the other two traffic scenarios. Overall, the proposed HCAGCN achieves the best prediction results among all models. Compared to that of HCAGCN, the total average ADE/FDE are decreased by 14.3%/12.5%, 14.9%/6.7%, 25.1%/23.3%, 35.7%/33.4%, 33.3%/30.9% and 48.5%/45.1% for EvolveGraph, Social-WaGDAT, VectorNet, CNN-LSTM, CS-LSTM and ARIMA, respectively. To intuitively display the prediction performance of HCAGCN, some example visualizations on ground-truth and predicted trajectories are shown in Fig. 7. It can be observed that the predicted trajectories are extremely close to the ground-truth trajectories in all scenarios.

### *D. Ablation Studies*

To further assess the effects of different model components on prediction accuracy, an ablation experiment is designed in which three variants are created by replacing modules in HCAGCN.

**HCAGCN-I***: In terms of the individual context extraction, TCN is replaced by LSTM to allow us to understand the contribution of the TCN to the final prediction.

**HCAGCN-S***: Regarding the scene context extraction, the paradigm of CNN with attention is substituted by a local CNN developed in [43]. This variant is aimed at evaluating the role played by the temporal attention in the full-fledged HCAGCN.

**HCAGCN-H***: In this variant, the STDGCN invented for interaction context extraction is replaced by the EvolveGraph to verify the superiority of STDGCN.

Table III presents the prediction results of the ablation experiment on INTERACTION dataset. In general, HCAGCN achieves the best performance in all scenarios. Comparing HCAGCN with HCAGCN-I*, it is obvious that TCN is more powerful in capturing the trajectory context than LSTM due to the ability of long-range memory is not suitable for the task of short-term features extraction in a connected environment. Besides, the better results produced by HCAGCN over HCAGCN-S* indicate that temporal attention is vital for the expression of scene context. The underlying reason is that temporal attention assigns different weights to the snapshots of scene image stream and thus the effects of scene images at different time intervals are accounted for. In addition, it can be found that HCAGCN outperforms HCAGCN-H*, which demonstrates that the proposed STDGCN has the superiority in modeling the spatio-temporal dynamics of the connected graph.

In real-world situations, the state of the target vehicle varies dynamically, which imposes a big challenge on the accurate prediction of vehicle trajectory. Hence, the performance of HCAGCN and its variants over various target vehicle states should be investigated. Here, target vehicle states are classified into three categories, namely uniform velocity, acceleration and deceleration. The specific classification rule is defined as follows according to [53]:

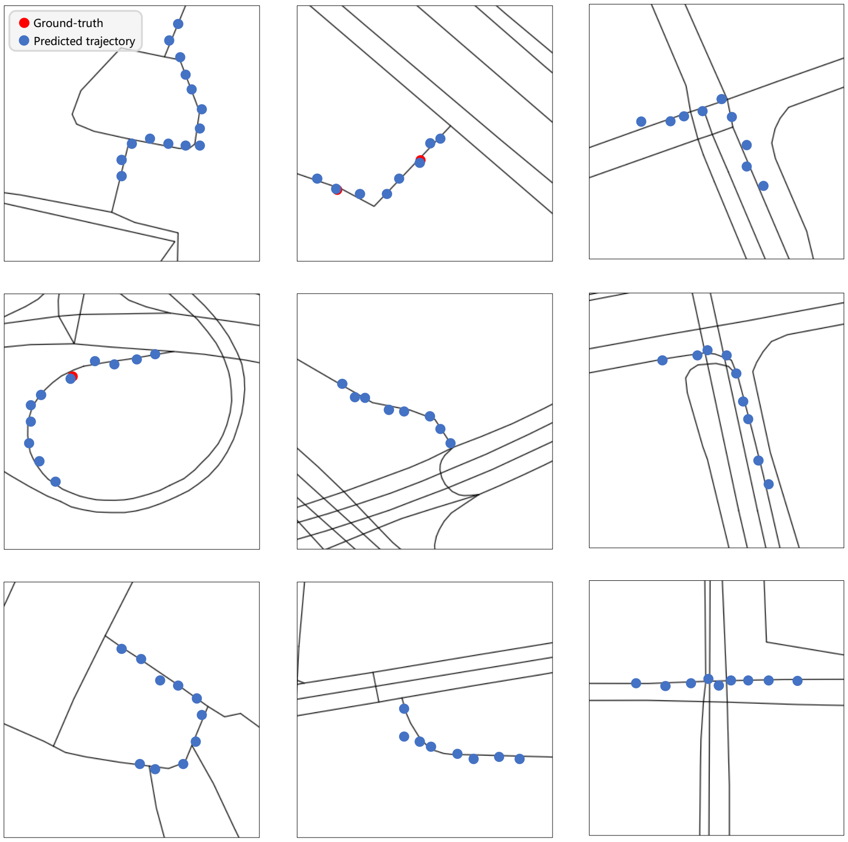
$$\mathrm{veh}_{\mathrm{state}} = \begin{cases} \text{acceleration}, & \eta \geq 1 \\ \text{uniform velocity}, & \eta \in (-1, 1) \\ \text{deceleration}, & \eta \leq -1 \end{cases} \tag{19}$$

The unit of $\eta$ is $\mathrm{m/s^2}$. Table IV exhibits the prediction results of different variants over various target vehicle states. Obviously, when the target vehicle travels at a uniform velocity, the gaps between the prediction errors generated by different variants are small. However, comparing HCAGCN-S* with HCAGCN, the gaps of ADE/FDE considerably increase from 0.03/0.09 to 0.09/0.26 and 0.10/0.30 over acceleration state and deceleration state, respectively. The same findings can be observed towards the comparison of HCAGCN and HCAGCN-I*. This proves the contributions of TCN and temporal attention to the stability of model performance. Furthermore, HCAGCN also yields better results over HCAGCN-H* for all kinds of vehicle states. This advances the confirmation of the superiority and robustness of STDGCN. The visualizations of prediction results given by HCAGCN over different vehicle states (see the Fig. 8) also justify the effectiveness of components ensemble.

As known, the online prediction of vehicle trajectory is vital and practical for incident prevention and traffic management. Therefore, the computational efficiency of applications of HCAGCN in real-time prediction should be assessed. Table V presents the average computational time of the proposed HCAGCN and its variants. It can be observed that HCAGCN family are able to perform the online trajectory prediction in the order of centisecond which is beneficial to the derived

TABLE II
PREDICTION RESULTS OF DIFFERENT MODELS IN VARIOUS SCENARIOS.

| Model | Roundabout | | Highway Ramp | | U-Intersection | |
|---|---|---|---|---|---|---|
| | ADE (m) | FDE (m) | ADE (m) | FDE (m) | ADE (m) | FDE (m) |
| ARIMA | 0.30 | 0.92 | 0.38 | 1.10 | 0.33 | 1.02 |
| CS-LSTM | 0.24 | 0.75 | 0.30 | 0.91 | 0.26 | 0.79 |
| CNN-LSTM | 0.26 | 0.81 | 0.32 | 0.95 | 0.25 | 0.77 |
| VectorNet | 0.23 | 0.70 | 0.27 | 0.83 | 0.22 | 0.67 |
| Social-WaGDAT | 0.20 | 0.62 | 0.22 | 0.67 | 0.19 | **0.52** |
| EvolveGraph | 0.22 | 0.67 | 0.21 | **0.60** | 0.21 | 0.64 |
| **HCAGCN** | **0.19** | **0.60** | **0.20** | 0.61 | **0.17** | 0.53 |



(a) Roundabout. (b) Highway Ramp. (c) Un-signalized Intersection.

Fig. 7. Visualization results over the three different scenarios. Black line corresponds to lane, red dot corresponds to ground-truth track point and blue dot corresponds to predicted track point.

TABLE III
PREDICTION RESULTS OF HCAGCN AND ITS VARIANTS IN VARIOUS SCENARIOS.

| Model | Roundabout | | Highway Ramp | | U-Intersection | |
|---|---|---|---|---|---|---|
| | ADE (m) | FDE (m) | ADE (m) | FDE (m) | ADE (m) | FDE (m) |
| HCAGCN-I* | 0.24 | 0.72 | 0.25 | 0.73 | 0.24 | 0.73 |
| HCAGCN-S* | 0.23 | 0.71 | 0.24 | 0.70 | 0.26 | 0.78 |
| HCAGCN-H* | 0.22 | 0.68 | 0.24 | 0.71 | 0.23 | 0.70 |
| **HCAGCN** | 0.19 | 0.60 | 0.20 | 0.61 | 0.17 | 0.53 |

services for intelligent transportation systems. Moreover, it should be noted that the average computational time is stable across various traffic scenarios, which further justifies the robustness of the proposed architecture.

## VI. Conclusion

In this study, the fundamental problem about the trajectory prediction of vehicles in a connected environment is revisited. The Heterogeneous Context-Aware Graph Convolutional Networks is proposed to extract individual context, scene context, and interaction context from historical states of connected vehicles and their driving environments. The core component of the proposed model is Spatio-Temporal Dynamic Graph Convolutional Network, which exploits both the spatial and temporal evolutions of interactional patterns and captures the high fidelity interaction context. Experimental results indicate that the proposed model achieves the best prediction performance over five stat-of-the-art methods in all types of driving scenarios. An ablation study further verifies the superiority and robustness of the proposed STDGCN when vehicles travel at fluctuating speeds.

In the future studies, it is worth introducing more abundant infrastructure information into the model development, such as traffic signs, road markings, and traffic lights. On the other hand, this work could be extended to tackle the multi-modality behaviors prediction considering the interactions between vehicles and pedestrians.

## Acknowledgment

This work is funded by the National Key R&D Program of China (2020YFB1600400).

## References

[1] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 296–305, 2013.

[2] D. Milakis, B. Van Arem, and B. Van Wee, "Policy and society related implications of automated driving: A review of literature and directions for future research," *J. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 324–348, 2017.

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

[4] S. Ammoun and F. Nashashibi, "Real time trajectory prediction for collision risk estimation between vehicles," in *Proc. IEEE 5th Int. Conf. Intell. Comput. Commun. Process.* IEEE, 2009, pp. 417–422.

[5] M. Schreier, V. Willert, and J. Adamy, "An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2751–2766, 2016.

[6] G. Xie, H. Gao, L. Qian, B. Huang, K. Li, and J. Wang, "Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using interactive multiple models," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5999–6008, 2017.

[7] H. Peng, B. Hu, Q. Shi, M. Ratcliffe, Q. Zhao, Y. Qi, and G. Gao, "Removal of ocular artifacts in eeg—an improved approach combining dwt and anc for portable applications," *IEEE J. Biomed. Health Inform.*, vol. 17, no. 3, pp. 600–607, 2013.

[8] X. Hu, J. Cheng, M. Zhou, B. Hu, X. Jiang, Y. Guo, K. Bai, and F. Wang, "Emotion-aware cognitive system in multi-channel cognitive radio ad hoc networks," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 180–187, 2018.

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

[10] X. Mo, Y. Xing, and C. Lv, "Interaction-aware trajectory prediction of connected vehicles using cnn-lstm networks," in *Proc. 46th Annu. Conf. IEEE Ind. Electron. Soc.* IEEE, 2020, pp. 5057–5062.

[11] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[12] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883–4894, 2019.

[13] W. Wang, J. Chen, Y. Zhang, Z. Gong, N. Kumar, and W. Wei, "A multi-graph convolutional network framework for tourist flow prediction," *ACM Trans. Internet Technol.*, vol. 21, no. 4, pp. 1–13, 2021.

[14] Y. Lu, H. Ding, S. Ji, N. Sze, and Z. He, "Dual attentive graph neural network for metro passenger flow prediction," *Neural. Comput. Appl.*, pp. 1–15, 2021.

[15] X. Li, X. Ying, and M. C. Chuah, "Grip: Graph-based interaction-aware trajectory prediction," in *Proc. IEEE 22nd Int. Conf. Intell. Transp. Syst. (ITSC)*, 2019, pp. 3960–3966.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[17] H. Jeon, J. Choi, and D. Kum, "Scale-net: Scalable vehicle trajectory prediction network under random number of interacting vehicles via edge-enhanced graph convolutional neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.* IEEE, 2020, pp. 2095–2102.

[18] X. Mo, Y. Xing, and C. Lv, "Heterogeneous edge-enhanced graph attention network for multi-agent trajectory prediction," *arXiv preprint arXiv:2106.07161*, 2021.

[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[20] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*. IEEE, 2017, pp. 399–404.

[21] P. Han, W. Wang, Q. Shi, and J. Yang, "Real-time short-term trajectory prediction based on gru neural network," in *IEEE/AIAA 38th Digit. Avion. Syst. Conf. (DASC)*. IEEE, 2019, pp. 1–8.

[22] S. Choi, J. Kim, and H. Yeo, "Attention-based recurrent neural network for urban vehicle trajectory prediction," *Procedia Comput. Sci.*, vol. 151, pp. 327–334, 2019.

[23] W. Chen, F. Wang, and H. Sun, "S2tnet: Spatio-temporal transformer networks for trajectory prediction in autonomous driving," in *Asian Conf. Mach. Learn.* PMLR, 2021, pp. 454–469.

[24] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recogn.*, 2019, pp. 12 126–12 134.

[25] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

[26] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: Graph convolutional neural networks with complex rational spectral filters," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 97–109, 2019.

[27] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[28] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

[29] X. Zhang, C. Huang, Y. Xu, L. Xia, P. Dai, L. Bo, J. Zhang, and Y. Zheng, "Traffic flow forecasting with spatial-temporal graph diffusion network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 17, 2021, pp. 15 008–15 015.

[30] G. Shen, X. Han, K. Chin, and X. Kong, "An attention-based digraph convolution network enabled framework for congestion recognition in three-dimensional road networks," *IEEE Trans. Intell. Transp. Syst.*, 2021.

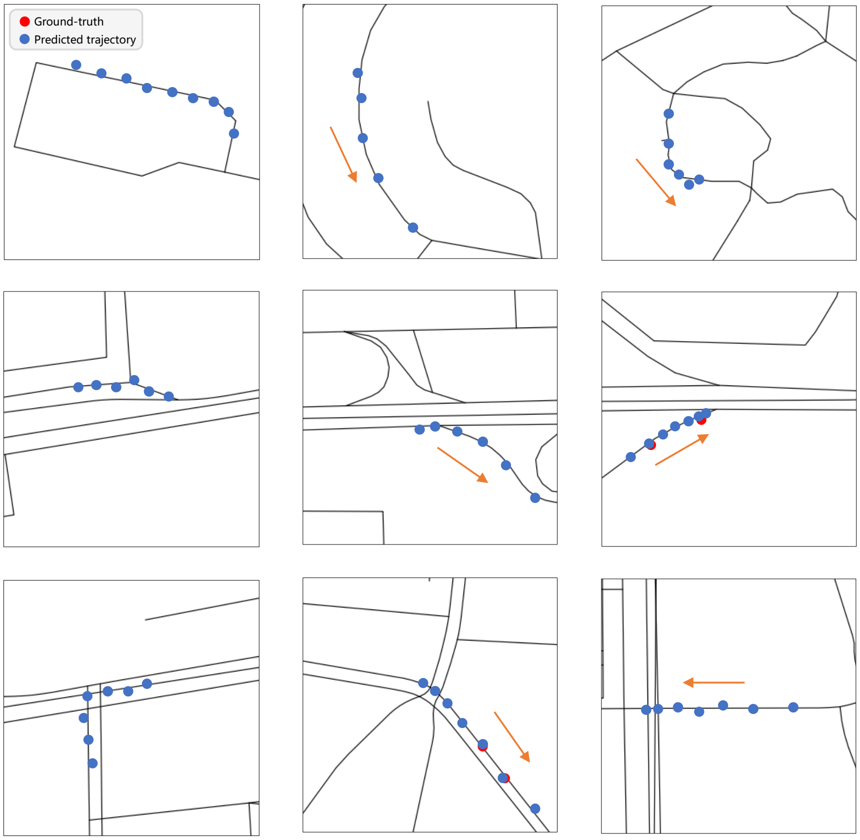
[31] M. Fang, L. Tang, X. Yang, Y. Chen, C. Li, and Q. Li, "Ftpg: A fine-grained traffic prediction method with graph attention network using big trace data," *IEEE Trans. Intell. Transp. Syst.*, 2021.

[32] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2020.

[33] R. Chandra, T. Guan, S. Panuganti, T. Mittal, U. Bhattacharya, A. Bera, and D. Manocha, "Forecasting trajectory and behavior of road-agents

TABLE IV
PREDICTION RESULTS OF HCAGCN AND ITS VARIANTS OVER VARIOUS STATES OF TARGET VEHICLE.

| Model | Uniform | | Acceleration | | Deceleration | |
|---|---|---|---|---|---|---|
| | ADE (m) | FDE (m) | ADE (m) | FDE (m) | ADE (m) | FDE (m) |
| HCAGCN-I* | 0.20 | 0.59 | 0.32 | 0.97 | 0.31 | 0.95 |
| HCAGCN-S* | 0.20 | 0.61 | 0.30 | 0.91 | 0.32 | 0.98 |
| HCAGCN-H* | 0.17 | 0.52 | 0.21 | 0.65 | 0.22 | 0.68 |
| **HCAGCN** | 0.16 | 0.49 | 0.19 | 0.59 | 0.19 | 0.60 |



(a) Uniform Speed. (b) Acceleration. (c) Deceleration.

Fig. 8. Visualization results over the three different vehicle states. The orange arrow indicates the travel direction. The results of first to third rows are produced in roundabout, highway ramp and un-signalized intersection scenarios respectively.

TABLE V
TIME COST OF DIFFERENT MODELS IN TESTING SET (MS).

| Model | Roundabout | Highway Ramp | U-Intersection |
|---|---|---|---|
| HCAGCN-I* | 69.910 | 70.141 | 69.442 |
| HCAGCN-S* | 66.003 | 65.209 | 65.012 |
| HCAGCN-H* | 68.487 | 68.481 | 67.149 |
| HCAGCN | 78.924 | 78.345 | 77.384 |

using spectral clustering in graph-lstms," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4882–4890, 2020.

[34] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recogn.*, 2020, pp. 11 525–11 533.

[35] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc.*, 2010, pp. 1045–1048.

[36] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Ling.*, 2016, pp. 207–212.

[37] D. Guo, S. Wang, Q. Tian, and M. Wang, "Dense temporal convolution network for sign language translation." in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 744–750.

[38] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, "Low latency acoustic modeling using temporal convolution and lstms," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 373–377, 2017.

[39] K. Zhang, Z. Liu, and L. Zheng, "Short-term prediction of passenger demand in multi-zone level: Temporal convolutional neural network with multi-task learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1480–1490, 2019.

[40] J. Li, H. Ma, Z. Zhang, J. Li, and M. Tomizuka, "Spatio-temporal graph dual-attention network for multi-agent prediction and tracking," *IEEE Trans. Intell. Transp. Syst.*, 2021.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2016,

pp. 770–778.

[42] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010.

[43] X. Mo, Y. Xing, and C. Lv, "Recog: A deep learning framework with heterogeneous graph for interaction-aware trajectory prediction," *arXiv preprint arXiv:2012.05032*, 2020.

[44] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recogn.*, 2019, pp. 3146–3154.

[45] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, "Learning to simulate complex physics with graph networks," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 8459–8468.

[46] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. Schardl, and C. Leiserson, "Evolvegcn: Evolving graph convolutional networks for dynamic graphs," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 04, 2020, pp. 5363–5370.

[47] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 802–810.

[48] W. Shu, K. Cai, and N. N. Xiong, "A short-term traffic flow prediction model based on an improved gate recurrent unit neural network," *IEEE Trans. Intell. Transp. Syst.*, 2021.

[49] C. Ma, C. Yang, F. Yang, Y. Zhuang, Z. Zhang, H. Jia, and X. Xie, "Trajectory factory: Tracklet cleaving and re-connection by deep siamese bi-gru for multiple object tracking," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*. IEEE, 2018, pp. 1–6.

[50] S. Haddad and S.-K. Lam, "Self-growing spatial graph networks for pedestrian trajectory prediction," in *Proc. IEEE Wint. Conf. Appl. Comput. Vis.*, 2020, pp. 1151–1159.

[51] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[52] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Ann. Stat.*, pp. 1189–1232, 2001.

[53] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle *et al.*, "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," *arXiv preprint arXiv:1910.03088*, 2019.

[54] J. Colyar and J. Halkias, "Us highway 101 dataset," *FHWA Tech. Rep.*, pp. 27–69, 2007.

[55] 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 Recogn.*, 2020, pp. 14 424–14 432.

[56] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8026–8037.

[57] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *arXiv preprint arXiv:1903.02428*, 2019.

[58] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Wint. Conf. Appl. Comput. Vis. (WACV)*. IEEE, 2017, pp. 464–472.

[59] B. M. Williams, P. K. Durvasula, and D. E. Brown, "Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transp. Res. Rec.*, vol. 1644, no. 1, pp. 132–141, 1998.

[60] J. Li, H. Ma, Z. Zhang, and M. Tomizuka, "Social-wagdat: Interaction-aware trajectory prediction via wasserstein graph double-attention network," *arXiv preprint arXiv:2002.06241*, 2020.

[61] J. Li, F. Yang, M. Tomizuka, and C. Choi, "Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020.

**Yuhuan Lu** received the B.Eng. and M.S. degrees in transportation engineering from Sun Yat-Sen University, Guangzhou, China, in 2017 and 2020, respectively. He is currently a PhD student with Department of Computer and Information Science, University of Macau, Macao, and also a research assistant with School of Intelligent Systems Engineering, Sun Yat-Sen University, Guangzhou, China. His research interests lie in Graph Embedding, Urban Computing and Intelligent Transportation Systems.

**Wei Wang** is currently an Associate Professor with School of Intelligent Systems Engineering, Sun Yat-sen University, China. He had been the UM Macao Research Fellow at University of Macau, Macau SAR. He received PhD degree in software engineering from Dalian University of Technology in 2017. His research interests include computational social science, data mining, internet of things, and artificial intelligence.

**Xiping Hu** is a Professor in School of Information Science and Engineering, Lanzhou University, Lanzhou, China, and the School of Intelligent Systems Engineering, Sun Yat-Sen University, Guangzhou, Guangdong, China. He received the Ph.D. degree from the University of British Columbia, Vancouver, BC, Canada. Also, he is the co-founder and Chief Scientist of Erudite Education Group Limited, Hong Kong, a leading language learning mobile application company with over 100 million users, and listed as top 2 language education platform globally. His research interests include mobile cyberphysical systems, crowdsensing, social networks, and cloud computing. He has more than 120 papers published and presented in prestigious conferences and journals. He has been serving as the lead Guest Editor of IEEE Transactions on Automation Science and Engineering, WCMC, IEEE Internet of Things Journal, etc.

**Pengpeng Xu** is an associate professor in the School of Civil Engineering and Transportation, South China University of Technology. He earned the BSc degree from Wuhan University of Technology, the MSc degree from Central South University, and the PhD degree from the University of Hong Kong. His research focuses primarily on road safety, big data, statistical learning, Bayesian inference, and spatial analysis, with particular interests in advancing traditional analytics by fusing multi-source data and by leveraging physics-based and data-driven methods. He now serves as the editorial board member of Accident Analysis & Prevention, Frontiers in Future Transportation, and Journal of Railway Science and Engineering. He was also invited as the guest editor of Sustainability and have completed more than 200 reviewer assignments for nearly 40 journals, thus being awarded as the Global Top Peer Reviewer by Web of Science.

**Shengwei Zhou** is currently a PhD student in the State Key Laboratory in the Internet of Things for Smart City (IOTSC) at the University of Macau. He received the BEng degree from Wuhan University in 2019, and the MSc degree from CUSP London at King's College London in 2020. He is broadly interested in theoretical computer science and urban informatics. His research interests span various topics in fair allocation problems and urban big data analytics.

**Ming Cai** received the B.E. degree from Sun Yat-sen University, Guangzhou, China, in 1999 and the Ph.D degree from Sun Yat-sen University, Guangzhou, China, in 2004. He is currently a full-time professor at School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou, China. His main research interests are in the traffic big data, traffic environmental engineering and intelligent transportation system.