Traffic Speed Prediction Based on Spatial-Temporal Dynamic and Static Graph Convolutional Recurrent Network

YANG Wenxi, WANG Ziling, CUI Tao, LU Yudong, QU Zhijian
School of Computer Science and Technology, Shandong University of Technology, Country Zibo, China

***Abstract*—Traffic speed prediction based on spatial-temporal data plays an important role in intelligent transportation. The time-varying dynamic spatial relationship and complex spatial-temporal dependence are still important problems to be considered in traffic prediction. In response to existing problems, a Dynamic and Static Graph Convolutional Recurrent Network (DASGCRN) model for traffic speed prediction is proposed to capture the spatial-temporal correlation in the road network. DASGCRN consists of Spatial Correlation Extraction Module (SCEM), Dynamic Graph Construction Module (DGCM), Dynamic Graph Convolution Recurrent Module (DGCRM) and residual decomposition. Firstly, the improved traditional static adjacency matrix captures the relationship between each time step node. Secondly, the graph convolution captures the overall spatial information between the road networks, and the dynamic graph isomorphic network captures the hidden dynamic dependencies between adjacent time series. Thirdly, spatial-temporal correlation of traffic data is captured based on dynamic graph convolution and gated recurrent unit. Finally, the residual mechanism and the phased learning strategy are introduced to enhance the performance of DASGCRN. We conducted extensive experiments on two real-world traffic speed datasets, and the experimental results show that the performance of DASGCRN is significantly better than all baselines.**

***Keywords*—Intelligent transportation; traffic speed prediction; spatial-temporal correlation; dynamic graph; graph convolution recurrent network**

I. INTRODUCTION

Traffic prediction is a crucial component of intelligent transportation systems, aiming to forecast future traffic conditions based on historical observational data. The systems optimize the use of traffic resource and enhance efficiency [1]. Traditional univariate time series prediction methods, including Historical Average (HA) [2], Vector Auto-Regression (VAR) [3], Support Vector Regression (SVR) [4] and Auto-Regressive Integrated Moving Average (ARIMA) [5], mainly focus on time dependence. There methods ignore the spatial-temporal correlation between nodes, resulting in poor prediction accuracy. Spatial-temporal feature fusion models involve data modeling across spatial and temporal dimensions, and have been widely used to solve traffic prediction challenges due to their significant versatility [6, 7].

In recent years, spatial-temporal graph neural networks have attracted the attention of researchers in the field of traffic prediction because of their excellent performance. Spatial-temporal graph neural network is a typical application of Graph Convolution Network (GCN) in spatial-temporal domain. At present, most popular traffic prediction methods are based on GCN, which uses predefined graph structure to capture spatial features between nodes, and uses Convolutional Neural Networks (CNN) [8] or Recurrent Neural Networks (RNN) [9,10] to extract temporal features [11, 12]. However, these methods rely heavily on predefined static graph structures, which directly affect the predictive performance of the model. To address the limitations of static graph structures, researchers have proposed a data-driven approach to adaptively generate adjacency matrices [13]. These methods improve the performance of the model by learning the parameters of the adjacency matrix in the training process and then calculating the similarity between the embeddings of learnable nodes [14].

However, the traffic prediction data show strong dynamic spatial-temporal correlation. The adjacency matrix generated by static adaptation is difficult to capture the complex dynamic characteristics of the road network. Researchers are increasingly interested in modeling dynamic nonlinear spatial-temporal correlations inherent in traffic data. Recent literature on STG-NCDE [15] combines adaptive graphs with neural controlled differential equations to further improve the performance of the model. STGODE [16] captures the dynamic spatial-temporal correlation of traffic flow through tensor-based ordinary differential equations, and combines semantic adjacency matrices and time-extended convolutional structures to capture long-distance spatial-temporal correlation. ST-GDN [17] uses multi-resolution traffic transformation information and local-global regional dependencies for prediction. DSTAGNN [18] directly mines historical traffic flow data to extract spatial-temporal correlations, and effectively capture the dynamic attributes of spatial associations between nodes. Traffic flow probability graphs [19] employ reinforcement learning to generate dynamic graph for extracting spatial-temporal features. DAGCRN [20] captures the spatial-temporal dependencies in traffic data through dynamic adjacency matrix and graph convolutional recurrent network, combined with spatial-temporal relationship extraction, adjacency matrix update and global temporal attention module. MHSRN [21] captures the moving features between timestamps through a hybrid convolution module, and designs a spatially aware multi-attention module to capture global and local spatial-temporal features. DSTGRNN [22] captures the spatial-temporal dependence in traffic data through the dynamic graph convolution module and generator, combines spatial-temporal relationship extraction, node embedding and dynamic feature

update, and integrates dynamic and static features to improve the accuracy of traffic flow prediction.

Considering the complex spatial-temporal correlation of traffic data, great progress has been made in traffic forecasting. However, there are still some challenges to be solved in the integration of dynamic spatial-temporal features. Firstly, the predefined adjacency matrix and adaptive adjacency matrix, which reflect the static structure of the traffic network, are static and cannot capture the dynamic characteristics of the actual traffic network over time. Therefore, this highlights the need for more sophisticated modeling of dynamic features in traffic prediction topologies. Secondly, the static distance graph and the dynamic attribute graph provide different perspectives on the topology of the traffic network. Their effective integration can provide more comprehensive and accurate topological information, to better capture spatial dependencies. Finally, most traffic forecasting methods generally do not distinguish between normal and abnormal traffic conditions, despite the prevalence of unexpected events such as accidents and traffic control measures. Therefore, a more in-depth exploration of anomalous traffic patterns is essential for enhancing the accuracy of traffic predictions.

In response to the identified challenges, a traffic speed prediction model that integrates spatial-temporal dynamic-static graph convolutional recurrent networks (DASGCRN) is proposed. Firstly, the learning aid self-recurrent unit matrix and trainable parametric matrix are added to the traditional adjacency matrix by the Spatial Correlation Extraction Module (SCEM). The pre-defined adjacency matrix is reconstructed to enhance the expressiveness of the traffic graph. Secondly, based on SCEM, the Dynamic Graph Construction Module (DGCM) introduces a hypernetwork based on gate control mechanism and sparse connection layer to construct dynamic adjacency matrices. Through dynamic node feature propagation, DGCM improves the stability and effectiveness of graph structures in dynamic environments. In addition, a dynamic graph generation method is proposed to capture the correlation between nodes while considering the periodic and dynamic changes of the traffic network. Furthermore, a dynamic graph convolutional recurrent model (DGCRM) based on RNNs integrates static and dynamic graphs to capture the spatial-temporal dependencies in traffic networks. Finally, in order to further improve the prediction performance of DASGCRN, a residual mechanism and a phased learning strategy are introduced.

The rest of this paper is organized as follows. The traffic prediction problem is formulated in Section II. Motivated by the challenges, we introduce the details of our solutions in Section III. After that, we evaluate our model by two real-world traffic datasets and derive the parameter studies and experimental results in Section IV. Studying the ablation experiments in Section V. Finally, we conclude our paper in Section VI.

II. Problem Formulation

Based on the connectivity of the actual road network among sensors in the given datasets, a corresponding road network topology graph $G = (V, E, A)$ can be generated. $V = \{v_1, \dots, v_N\}$ represents the set of all nodes in the network and N denotes the total number of nodes, with each node representing a traffic sensor deployed at the roadside responsible for recording traffic information at its location. E denotes the set of edges that represents the spatial connectivity between nodes. The adjacency matrix $A \in R^{N \times N}$ is employed to depict the spatial adjacency relationships between nodes, where a_{ij} is an element of matrix A indicating the spatial connection status between nodes v_i and v_j. If $v_i, v_j \in V$ is connected to $(v_i, v_j) \in E$, then $a_{ij} = 1$; otherwise $a_{ij} = 0$. The relationship between road connectivity and the road network topology is illustrated in Fig. 1.

In this context, $X_t = [x_t^1, \dots, x_t^N]^T \in R^{N \times L \times C}$ denote the traffic speed at the N nodes of topology graph G at time t, L represent the total length of each time series, and C indicate the total number of node feature types. The traffic speed prediction problem is defined as shown in Eq. (1): Given the speed sequence $X_{1:P} = [X_1, \dots, X_P]^T \in R^{P \times N \times L}$ and topology graph G, the objective is to learn a mapping function F to predict the future sequence $\tilde{X}_{(P+1)(P+Q)} = \left[\tilde{X}_{(P+1)}, \dots, \tilde{X}_{(P+Q)}\right]^T \in R^{Q \times N \times L}$. Here, P represents the length of the historical speed data sequence, and Q denotes the length of the speed sequence to be predicted.

$$[X_{1:P}, G] \xrightarrow{F} \tilde{X}_{(P+1)(P+Q)} \tag{1}$$

III. The Proposed DASGCRN Model

A. Model Architecture

The architecture and various modules of the proposed DASGCRN model are illustrated in Fig. 2. The model consists of four main modules: Spatial Correlation Extraction Module (SCEM), Dynamic Graph Construction Module (DGCM), Dynamic Graph Construction Recurrent Module (DGCRM), Residual Decomposition and Model Training Strategy. SCEM models the spatial relationship between nodes based on static adjacency matrix. DGCM module is responsible for modeling the dynamic spatial-temporal relationship between nodes and edges, effectively capturing the potential dynamic correlation in the traffic network, and generating the dynamic adjacency matrix. The DGCRM module integrates dynamic and static adjacency matrices to model long-term dependencies between historical and future time steps by focusing on connections between different nodes in the transportation network, capturing time dependencies from a global perspective. In addition, residual traffic improves the training process of the model by adding skip connections between different layers of the network. Finally, piecewise learning training strategy is used to promote model convergence.

B. Spatial Correlation Extraction Module (SCEM)

1) In traffic road networks, traditional static adjacency matrices are typically used to represent the static connectivity relationships between nodes, as shown in Eq. (2). Where A_{v_i,v_j} denotes the edge weight between sensors v_i and v_j, d_{v_i,v_j} represents the road network distance from node v_i to v_j. Additionally, σ indicates the standard deviation of the distance, and κ denotes the sparsity threshold.

$$A_{v_i,v_j} = \begin{cases} exp\left(-\frac{d_{v_i,v_j}}{\sigma^2}\right), v_i \neq v_j, d_{v_i,v_j} \leq \kappa \\ 0, otherwise \end{cases} \tag{2}$$

2) The static adjacency matrix reflects the static connection state of the road. While the central road node is restricted to processing only the information from neighboring nodes, thus ignoring the data from distant nodes with similar traffic flow patterns [23]. The static adjacency matrix only considers the spatial distance factor, ignoring the influence of time factor on the relationship between nodes. In the real traffic road network, the spatial relationship between nodes is affected by time change [24], which limits the representation of dynamic spatial relationship in the traditional static adjacency matrix. Inspired by DAGCRN [20], the improved static adjacency matrix incorporates an identity matrix and a parametric matrix, as shown in Eq. (3).

$$\tilde{A} = A + A_{par} + I_N \tag{3}$$

3) where $\tilde{A}$ represents the improved static adjacency matrix, A denotes the traditional static adjacency matrix, and I_N is the identity matrix with diagonal elements equal to 1, indicates that each node is connected to itself (self-recurrent). This mechanism.

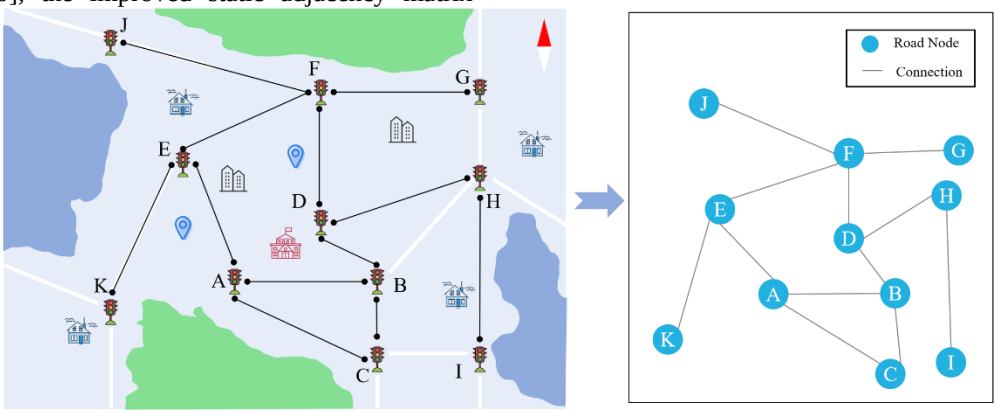

Fig. 1. The corresponding relationship between road connection and road network topology diagram.

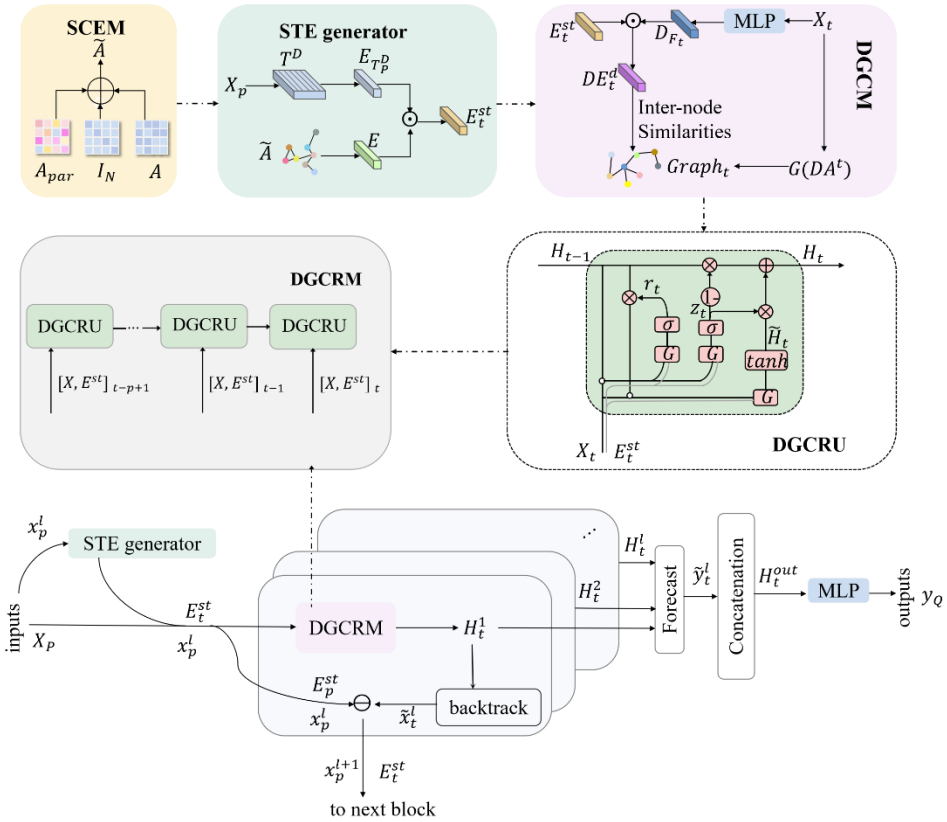

Fig. 2. The architecture and modules of DASGCRN.

4) It allows nodes to retain their own information while considering the impact of their state on neighboring nodes at each update step. The goal is to enhance the model's ability to capture long-term dependencies related to changes in traffic speed variations. A_{par} is the parametric matrix, initialized randomly and optimized through continuous iterations. As shown in Fig. 3, A_{par} enables the dynamic learning of the weight relationships between the central node (pink pentagon) and surrounding nodes (brown and beige circles). Allows adaptive adjustment of the connection based on these weights. This flexibility helps to explore the dynamic dependence of traffic speed over time in static network space. After A_{par} is optimized across all training samples, its combination with A and I_N forms $\tilde{A}$, enabling $\tilde{A}$ to more accurately represent the dynamic spatial relationships in the traffic network.

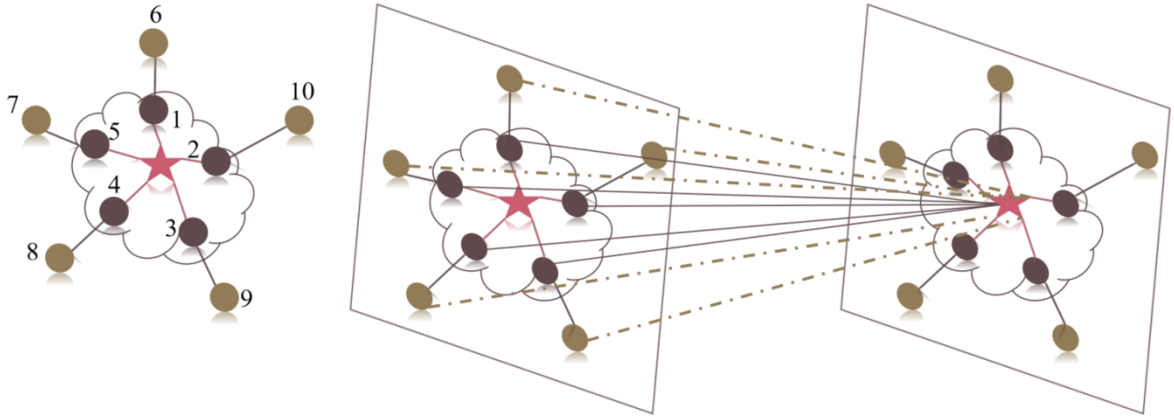

Fig. 3. Parametric matrix diagram.

C. Dynamic Graph Construction Module (DGCM)

The Dynamic Graph Construction Module (DGCM) has been carefully designed to efficiently capture spatial dependencies in traffic speed sequences, as well as temporal variations in these sequences. Specifically, the algorithm first builds a Spatial-Temporal Embedding generator (STE generator), which consists of two basic components: a spatial embedding generator and a temporal embedding generator. These components work together to extract and represent the spatial and temporal features present in the traffic speed sequence, thereby improving the overall understanding of its dynamics.

The core function of the spatial embedding generator is to learn the graph structure from input $\tilde{A}$, after which the learned node information is processed through two fully connected layers to produce the spatial embedding E. The primary task of the temporal embedding generator is to determine the daily encoding embedding $E_{T_P^D}$ that corresponds to the current traffic speed sequence X_P. Subsequently, $E_{T_P^D}$ and E are combined using an element-wise multiplication operation to obtain the new spatial-temporal embedding E_t^{st}. The implementation process of the spatial-temporal embedding is shown in Eq. (4).

$$E_t^{st} = E_{T_P^D} \odot E \tag{4}$$

where $E_{T_P^D} \in R^{P \times N \times D}$ is the daily embedding composed of consecutive time steps $P = [t - p + 1, \dots, t]$, and the p denotes the input sequence length. Additionally, $\odot$ signifies the element-wise multiplication operation. The input at the current time step x_t is processed through a multi-layer perceptron MLP to extract dynamic features, as shown in Eq. (5).

$$D_{F_t} = MLP(x_t) \tag{5}$$

where, $D_{F_t} \in R^{N \times D}$ represents the dynamic features of the current time step x_t after transformation. An element-wise multiplication operation is performed between D_{F_t} and E_t^{st}, and the resulting product is normalized to generate the dynamic graph embedding DE_t^d, as shown in Eq. (6).

$$DE_t^d = tanh\left(D_{F_t} \odot E_t^{st}\right) \tag{6}$$

The dynamic adjacency matrix DA^t at time t is calculated using the similarity between nodes, as shown in Eq. (7). This dynamic adjacency matrix expresses the evolving connectivity and changing trends among the network nodes. Furthermore, α is a hyperparameter used to control the saturation rate of the activation function.

$$DA^t = ReLU\left(tanh\left(\alpha\left(DE^d DE^{d^T}\right)\right)\right) \tag{7}$$

D. Dynamic Graph Convolutional Recurrent Module (DGCRM)

The Dynamic Graph Convolutional Recurrent Module (DGCRM) primarily integrates the improved static adjacency matrix $\tilde{A}$ with the generated dynamic adjacency matrix DA^t. This integration aggregates the information between the nodes of the traffic network and their neighbors, and effectively captures the dynamic spatial-temporal dependencies in the traffic network. Both $\tilde{A}$ and DA^t reflect the correlations among nodes from different perspectives. The DGCRM module effectively combines these matrices to provide a comprehensive view of the traffic road network for predictive modeling, accommodating the dynamic characteristics of the graph structure over time, as detailed in Eq. (8) to Eq. (12).

DGCRM utilizes X_P, $\tilde{A}$, and DA^t as inputs for the graph convolutional layer, and computes a weighted average to the outputs of the graph convolution.

$$\widetilde{H}_t^{(k)} = \alpha H_t^{in} + \beta H_t^{(k-1)} DA^t + \gamma H_t^{(k-1)} \tilde{A} \quad (8)$$

$$H_t^{(k)} = ReLU\left(\left(\mu H_t^{in} + (1-\mu)\widetilde{H}_t^{(k)}\right) W_P^{(k)}\right) \quad (9)$$

$$H_t^{out} = \sum_{i=0}^{K} H_t^{(k)} W^{(k)} \quad (10)$$

$$H_t^{(0)} = H_t^{in} \quad (11)$$

$$H_t^{out} = \Theta_{*G}\left(H_t^{in}, DA^t, \tilde{A}\right) \quad (12)$$

where α, β and γ are hyperparameters that adjust the weights of different components. $W_P^{(k)}$ denotes the k-th order parameter matrix. Hyperparameter μ is used to control the retention rate of the original node information in the process of information transmission, and deeper neighborhood exploration is carried out while preserving the local structure. H_t^{in} and H_t^{out} represent the input and output node states of the dynamic graph convolution at time step t, respectively. $W^{(k)} \in R^{k \times d_{in} \times d_{out}}$ is the learnable feature transformation matrix, and k denotes the model propagation depth. $H_t^{out} = \Theta_{*G}\left(H_t^{in}, DA^t, \tilde{A}\right)$ serves as a simplified representation of the dynamic graph convolution process, where Θ indicates dynamic graph generation, and $*G$ represents dynamic graph convolution. In the context of the DGCRU module diagram, G is used to denote dynamic graph convolution.

DGCRM is composed of Dynamic Graph Convolutional Gated Recurrent Unit (DGCRU), with the hidden state H_t^l of the final DGCRU serving as the output of the DGCRM. DGCRU is formulated by substituting the matrix multiplication operations in the GRU with dynamic graph convolution modules, as detailed in Eq. (13) to Eq. (16).

$$z_t = \sigma(\Theta_{*G}^z [X_t || H_{t-1}, E_t^{st}]) \quad (13)$$

$$r_t = \sigma(\Theta_{*G}^r [X_t || H_{t-1}, E_t^{st}]) \quad (14)$$

$$\widetilde{H}_t = tanh\left(\Theta_{*G}^h [X_t || r_t \odot H_{t-1}, E_t^{st}]\right) \quad (15)$$

$$H_t = (1 - z_t) \odot H_{t-1} + z_t \odot \widetilde{H}_t \quad (16)$$

where X_t and H_t represent the input and output at time step t, respectively. The symbol $\odot$ denotes the Hadamard product, while $||$ indicates concatenation operations. The function $\sigma(\cdot)$ refers to the sigmoid activation function. z_t and r_t are the update gate and reset gate at time t. Respectively, $\widetilde{H}_t$ represents the candidate state of the GRU unit. The symbol $*G$ signifies dynamic graph convolution, and Θ^z, Θ^h, and Θ^r correspond to the learnable parameters associated with the respective graph convolutions.

E. Residual Decomposition

To achieve multi-step predictions and sequence decomposition of traffic flow. An output sublayer consisting of linear layers is constructed after DGCRM. The mathematical expressions for the output sublayer are shown in Eq. (17) and Eq. (18).

$$\tilde{y}_t = Linear_{1:t}\left(H_t^l\right) \quad (17)$$

$$\tilde{x}_t = Linear_{1:t}\left(H_t^l\right) \quad (18)$$

where the predicted output $\tilde{y}_t^l \in R^{Q \times N \times C}$ denotes the prediction of y_t by the l-th block based on x_p^l, while the backward predicted output $\tilde{x}_t^l \in R^{P \times N \times C}$ represents the reverse prediction of x_p^l by the l-th block. The final output of the prediction model is given in Eq. (19) and Eq. (20).

$$y_Q = \sum_1^l \tilde{y}_t^l \quad (19)$$

$$x_p^{l+1} = x_p^l - \tilde{x}_t^l \quad (20)$$

Backward prediction can be understood as the decomposition process of traffic speed sequence. The aim is to eliminate the part of the information that the model has learned, while retaining the unlearned part for further processing by reconstructing the velocity sequence. Specifically, $\tilde{x}_t^l$ can be viewed as the reconstruction of the input speed sequence x_p^l, which incorporates the information that H_t^l has learned from x_p^l. By utilizing a residual decomposition mechanism, the learned information from x_p^l is removed, and the unlearned components of x_p^{l+1} are preserved for modeling in the subsequent block. The outputs from each block are then summed to produce the final prediction result.

F. Model Training

In order to improve the training efficiency of the model, a phased learning strategy is designed. In the initial phase of training, instead of training all blocks at the same time, only the first few blocks are trained. As the number of training layers increases, additional blocks are gradually incorporated into the training process. This approach significantly reduces the time and memory overhead required in the early stages of training. The training process is illustrated in Algorithm 1.

Algorithm 1: Training Algorithm of the DASGCRN Model.

Input: The static graph $G = (V, E, A)$, traffic speed sequence $X_p \in R^{P \times N \times D}$, current time step $X_t \in R^{T \times N \times D}$, time embedding T^D, spatial embedding E, training epochs $epochs$;

```
set turn = 1, i = 1
repeat
    initialize hidden state H_t^0, randomly select a batch (input X ∈ R^{P×N×D}, output Y ∈ R^{Q×N×D}, time of day T ∈ R^{Q×N×D}) from X_p;
    if turn%s == 0 and i < K then
        i = i + 1
    end if
        for p in 0,1,...,P − 1 do
            Calculate ỹ_t^l and x̃_t^l according to Eq.16, Eq.17 ;
    end for
        sum ỹ_Q = Σ_1^l ỹ_t^l
        Compute loss L = loss(ỹ_Q, y_Q)
        Back propagation and update parameters according to L;
        turn = turn + 1;
    Until the model reaches a stable state;
```

Output: the leaned DASGCRN model.

IV. EXPERIMENTS

A. Datasets

The experiment used two real-world traffic history datasets to verify the performance of different prediction models, as detailed in Table I.

- Los-loop Dataset

The dataset, from the City of Los Angeles, records speed information collected by loop detectors from March 1-7, 2012. It includes 207 road nodes and 1,313 road connections, collecting speed data every five minutes.

- SZ-taxi Dataset.
- The dataset comes from taxi tracks in Shenzhen City from January 1-31, 2015, covering 156 major roads in Luohu District, including 266 road connection relationships, with speed data collected every 15 minutes.

TABLE I. DETAILED INFORMATION OF TRAFFIC DATASETS FOR EXPERIMENT

Dataset	Number of Sensors	Edge	Time Interval	Time Range
SZ-taxi	156	266	15min	2015.1.1-2015.1.31
Los-loop	207	1313	5min	2012.3.1-2012.3.7

B. Parameter Settings

According existing methods, we split the datasets into training and testing sets in the same way as the baseline, i.e. 8:2 on the Los-loop and SZ-taxi datasets. Historical traffic speed data from the past hour is used to predict traffic speed in the future hour. The time step for the Los-loop dataset is set to 12 (sampling for 5 minutes), while the time step for the SZ-taxi dataset is set to 4 (sampling for 15 minutes). The experiment was repeated for 5 times and the average value of the evaluation index was reported.

The model employed the Adam optimizer with a learning rate of 0.003, 64 hidden units, an embedding dimension of 10, a batch size of 64, and an epoch value of 300. Mean Absolute Error (MAE) was used as the training loss function, along with an early stopping strategy to prevent overfitting.

C. Baseline Methods

HA: The Historical Average method (HA) predicts future speeds by analyzing the average values of historical speed data.

SVR: Support Vector Regression (SVR) is one of the classical time series analysis models that employs linear support vector machines to perform regression tasks.

Graph Convolutional Network (GCN): GCNs are capable of considering the spatial structural characteristics of graphs, extending convolution operations to graph structures.

Gated Recurrent Unit (GRU): GRUs can capture significant data dependencies over large gaps in time series.

T-GCN [11]: The T-GCN model simultaneously captures spatial and temporal dependencies using GCN and GRU.

A3T-GCN [25]: The A3T-GCN model introduces an attention mechanism to the T-GCN framework to extract weight information for each time step.

NA-DGRU [26]: The NA-DGRU model utilizes two GRUs to extract correlations between speed and time from both the original features and aggregated neighborhood features.

MTA-CN [27]: The MTA-CN model transforms long time single feature historical data into short time multi-feature data and incorporates a two-stage attention mechanism to capture the significance of features in different time periods and time steps.

FD-TGCN [28]: The spatial module introduces a novel Dynamic Convolution Matrix (DCM) to learn the characteristics of dynamic road structures, while the temporal module employs a Fast Temporal Convolution Network (FTCN) to model long-term temporal relationships.

D. Experimental Evaluation Metrics

The experiment employs five commonly used evaluation metrics to assess the performance of various prediction methods:

1) Mean Absolute Error (MAE): This index intuitively reflects the actual forecast error, as shown in Eq. (20).

2) Root Mean Squared Error (RMSE): RMSE is sensitive to outliers and is often used as a standard to measure the predictive performance of deep learning models, as shown in Eq. (21).

3) Accuracy (ACC): This measure describes the degree of fit between the predicted value and the true value; The closer the value is to 1, the better the prediction performance, as shown in Eq. (22).

4) Coefficient of Determination (R2): This statistic evaluates the goodness of fit of regression models; higher values indicate better predictive accuracy, as defined in Eq. (23).

5) Explained Variance Score (var): This score measures the model's ability to explain the variance in the data; values closer to 1 indicate a higher explanatory power of the model, as defined in Eq. (24).

$$MAE = \frac{1}{MN}\sum_{j=1}^{M}\sum_{i=1}^{N}\left|y_{ij} - \tilde{y}_{ij}\right| \tag{20}$$

$$RMSE = \sqrt{\frac{1}{MN}\sum_{j=1}^{M}\sum_{i=1}^{N}\left(y_{ij} - \tilde{y}_{ij}\right)^2} \tag{21}$$

$$ACC = 1 - \frac{Y - \tilde{Y}_F}{Y_F} \tag{22}$$

$$var = 1 - \frac{Var\{Y - \tilde{Y}\}}{Var\{Y\}} \tag{23}$$

$$R^2 = 1 - \frac{\sum_{j=1}^{M}\sum_{i=1}^{N}\left(y_{ij} - \tilde{y}_{ij}\right)^2}{\sum_{j=1}^{M}\sum_{i=1}^{N}\left(y_{ij} - \bar{Y}\right)^2} \tag{24}$$

In the above equations, y_{ij} and $\tilde{y}_{ij}$ represent the actual and predicted traffic speeds on road i-th at time t, respectively. M denotes the sample size of the traffic series, while N represents the set of roads. Y and $\tilde{Y}$ denote the sets of y_{ij} and $\tilde{y}_{ij}$, with $\bar{Y}$ being the average of Y. Both MAE and RMSE describe error

values, where smaller values indicate better model performance. On the contrary, the accuracy reflects the correctness of the prediction, and the higher the value, the better the prediction effect.

E. Experimental Results

Table II presents a comparison of the DASGCRN model with nine baseline models on the Los-loop and SZ-taxi datasets. The metrics underlined in the table indicate the best results, while the models in bold italic type denote those whose prediction results are directly referenced from the published papers. A "/" indicates that the baseline model did not provide that metric at the time of publication. It is evident from the experimental results that the proposed DASGCRN model outperforms all other models on all evaluation indicators of the two datasets. Statistical methods and traditional machine learning models have high requirements for data stationarity and usually focus only on temporal correlation. This leads to challenges in meeting these requirements for traffic data, resulting in poor predictive performance from traditional methods. In contrast, compared with GCN and GRU, the DASGCRN model improved on all evaluation measures. This shows that the model effectively captures the spatial topological features of urban road network and the temporal changes of traffic state. In addition, other baseline models also achieve better predictive performance compared to traditional methods, highlighting the strong spatial-temporal correlations in traffic data. Capturing these spatial-temporal features enhances traffic prediction accuracy.

TABLE II. COMPARISON DASGCRN AND BASELINE MODELS ON LOS-LOOP AND SZ-TAXI

T	**Datasets**	**Los-loop**									
	Methods	HA	SVR	GCN	GRU	***TGCN***	***A3T-GCN***	***NA-DGRU***	***MTA-GN***	***FD-TGCN***	DSAGCRN
15min	MAE	4.0162	3.7285	5.3525	3.0602	3.1802	3.1365	3.0281	3.1004	3.083	2.7888
	RMSE	7.5323	6.0084	7.7922	5.2182	5.1264	5.0904	5.1348	5.1058	5.133	5.0462
	ACC	0.8715	0.8977	0.8673	0.9109	0.9127	0.9133	0.9126	/	0.9119	0.9140
	R2	0.7083	0.8123	0.6843	0.8576	0.8634	0.8653	/	0.8670	/	0.8675
	Var	0.7084	0.8146	0.6844	0.8577	0.8634	0.8653	/	0.8679	/	0.8682
30min	MAE	4.4136	3.7248	5.6118	3.6505	3.7466	3.6610	3.6692	3.6041	3.712	3.0812
	RMSE	8.3204	6.9588	8.3353	6.2802	6.0598	5.9974	6.1358	5.8462	5.904	5.8436
	ACC	0.8581	0.8815	0.8581	0.8931	0.8968	0.8979	0.8955	/	0.8960	0.9004
	R2	0.6408	0.7492	0.6402	0.7957	0.8098	0.8173	/	0.8148	/	0.8205
	Var	0.6409	0.7523	0.6404	0.7958	0.8100	0.8173	/	0.8148	/	0.8220
45min	MAE	4.7898	4.1288	5.9534	4.0915	4.1158	4.1712	4.0567	3.9483	/	3.2735
	RMSE	9.0213	7.7504	8.8036	7.0343	6.7065	6.6840	6.7604	6.5731	/	6.3597
	ACC	0.8462	0.8680	0.8500	0.8801	0.8857	0.8861	0.8851	/	/	0.8916
	R2	0.5783	0.6899	0.5999	0.7446	0.7679	0.7694	/	0.7726	/	0.7869
	Var	0.5783	0.6947	0.6001	0.7451	0.7684	0.7705	/	0.7732	/	0.7894
60min	MAE	5.1504	4.5036	6.2892	4.5186	4.6021	4.2343	4.4256	4.0154	4.535	3.4260
	RMSE	9.6602	8.4388	9.2657	7.6621	7.2677	7.0990	7.2776	6.8749	6.983	6.7397
	ACC	0.8354	0.8562	0.8421	0.8694	0.8762	0.8790	0.8764	/	0.8760	0.8852
	R2	0.5070	0.6336	0.5583	0.6980	0.7283	0.7407	/	0.7492	/	0.7605
	Var	0.5070	0.5593	0.5593	0.6984	0.7290	0.7415	/	0.7495	/	0.7637
T	Datasets	SZ-taxi									
	Metric	HA	SVR	GCN	GRU	***TGCN***	***A3TGCN***	***NA-DGRU***	***MTA-CN***	***FD-TGCN***	DSAGCRN
15min	MAE	2.7842	2.6233	4.2367	2.5955	2.7145	2.6840	2.7387	2.6105	2.667	2.4964
	RMSE	4.2991	4.1455	5.6596	3.9994	3.9825	3.9989	4.0587	4.0440	4.036	3.9716
	ACC	0.7005	0.7012	0.6107	0.7149	0.7195	0.7218	0.7173	/	0.7187	0.7233
	R2	0.8305	0.8423	0.6654	0.8329	0.8539	0.8512	/	0.8526	/	0.8554
	Var	0.8305	0.8424	0.6655	0.8329	0.8539	0.8512	/	0.8530	/	0.8559
30min	MAE	2.8191	2.6875	4.2647	2.6906	2.7522	2.7038	2.7280	2.6158	2.778	2.5064
	RMSE	4.3508	4.1628	5.6918	4.0942	4.0317	4.1749	4.0683	4.0684	4.083	3.9899
	ACC	0.6969	0.7100	0.6085	0.7184	0.7167	0.7202	0.7166	/	0.7154	0.7221

	R2	0.8264	0.8410	0.6616	0.8249	0.8451	0.8493	/	0.8507	/	0.8541
	Var	0.8264	0.8413	0.6617	0.8250	0.8451	0.8493	/	0.8512	/	0.8546
45min	MAE	2.8488	2.7359	4.2844	2.7743	2.7645	2.7261	2.7393	2.6954	/	2.5157
	RMSE	4.3916	4.1885	5.7142	4.1534	4.0910	4.2461	4.0777	4.1168	/	4.0034
	ACC	0.6941	0.7082	0.6069	0.7143	0.7155	0.7186	0.7159	/	/	0.7211
	R2	0.8231	0.8391	0.6589	0.8198	0.8436	0.8474	/	0.8412	/	0.8531
	Var	0.8231	0.8397	0.6590	0.8199	0.8436	0.8474	/	0.8415	/	0.8536
60min	MAE	2.8754	2.7751	4.3034	2.7712	2.7860	2.7391	2.7487	2.6396	2.762	2.5254
	RMSE	4.4302	4.2156	5.7361	4.0747	4.1299	4.2707	4.0851	4.1637	4.104	4.0167
	ACC	0.6914	0.7063	0.6054	0.7197	0.7142	0.7169	0.7154	/	0.7141	0.7202
	R2	0.8199	0.8370	0.6564	0.8266	0.8421	0.8454	/	0.8434	/	0.8521
	Var	0.8199	0.8379	0.6564	0.8267	0.8421	0.8454	/	0.8440	/	0.8526

The experiment further divided the data of one day from the Los-loop and SZ-taxi test sets. It drew the prediction curves of DASGCRN and T-GCN models within the interval of 5 minutes and 60 minutes, and compared them with the Ground Truth, as shown in Fig. 4. As can be seen from the dotted box in Fig. 4, the DASGCRN model is more sensitive to capturing data with sudden increases or decreases in speed than the T-GCN, so the predictions are more accurate. This improved performance is attributed to the integration of time series modeling in DASGCRN's dynamic graph structure, which enables the model to focus on specific patterns of dynamic change, resulting in faster and more accurate predictions. As highlighted by the pink dotted boxes in Fig. 4(c) and Fig. 4(d), DASGCRN showed superior performance in both short (5 minutes) and long (60 minutes) predictions.

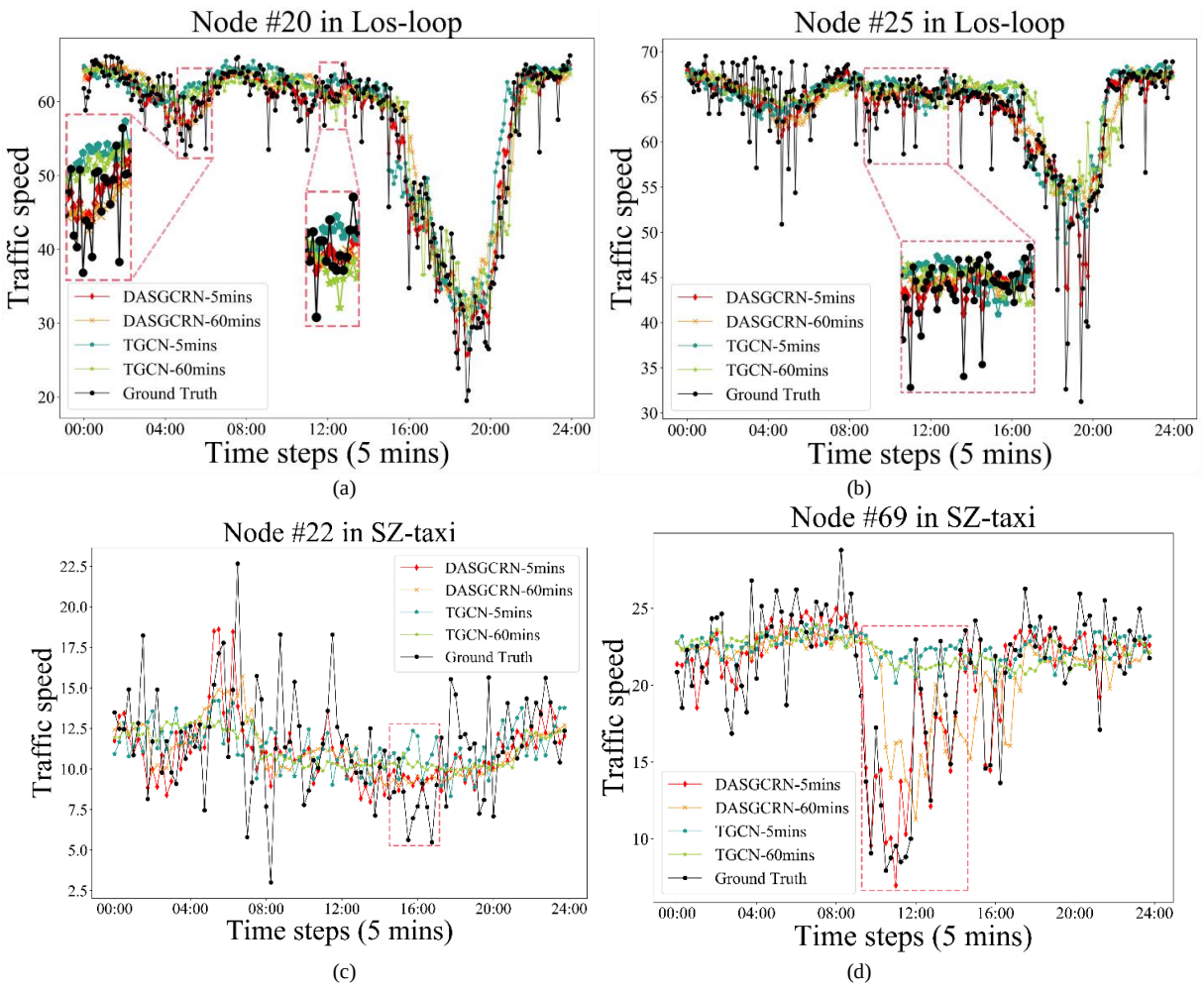

Fig. 4. The prediction curves of two datasets.

Fig. 5 illustrates the absolute error heatmaps of predicted values versus actual values for the DASGCRN model across two datasets. Due to the large size of the dataset, we selected only the first 60 time steps of the 12 roads in each dataset and generated heatmaps for each dataset with four different prediction ranges. Generally, the prediction performance of the model tends to decline with the increase of the prediction range. However, the heatmap shows that DASGCRN has maintained good performance across both the short-term and long-term forecast ranges. This is due to the dynamic graph convolution module used in the DASGCRN model, which replace the matrix multiplication in GRU. It can allow for more flexible control over feature information transmission and enhance the model's ability to capture long-term dependencies.

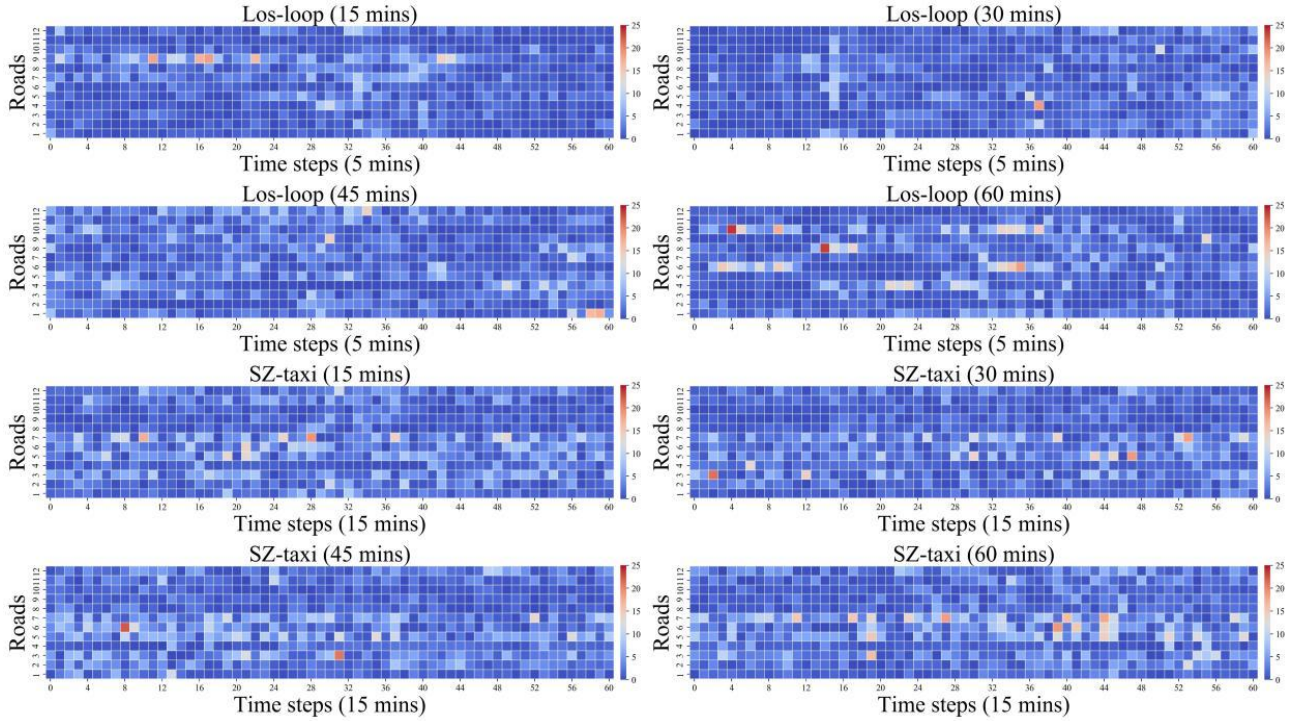

Fig. 5. The heatmaps of different forecasting granularity of DASGCRN two datasets.

V. Ablation Study

To validate the effectiveness of key components in the DASGCRN model, nine variants of the DASGCRN model were designed:

w/o I_N: DASGCRN without the self-loop adjacency matrix.

w/o A_{par}: DASGCRN without the learnable parameter matrix.

w/o DA_t: DASGCRN without the dynamic adjacency matrix.

w/o FCL: Replacing dynamic graph convolution with a simple fully connected layer in DASGCRN.

w/o T^D: DASGCRN without temporal embeddings, using only spatial embeddings to generate dynamic embeddings.

w/o DA_tI_N: DASGCRN without dynamic graph convolution and the self-loop adjacency matrix.

w/o DA_tA_{par}: DASGCRN without dynamic graph convolution and the learnable parameter matrix.

w/o Mix: DASGCRN without residual connections in dynamic graph convolution, using the last hop's output as the output of dynamic graph convolution.

w/o TS: DASGCRN without segmented training.

TABLE III. Comparison of Ablation Results of DASGCRN and Variant on Los-loop Dataset

T	Datasets	Los-loop									
	Methods	DASGCRN	w/o I_N	w/o A_{par}	w/o DA_t	w/o FCL	w/o T^D	w/o DA_tI_N	w/o DA_tA_{par}	w/o Mix	w/o TS
15min	MAE	2.7888	3.6346	3.6731	9.1303	7.7888	3.4029	11.5724	13.6317	3.4096	2.9895
	RMSE	5.0462	6.3577	6.0787	13.7764	11.0462	5.5153	15.6325	17.0738	5.4673	5.3043
	ACC	0.8715	0.8977	0.8673	0.9109	0.9127	0.9133	0.9126	/	0.9119	0.9140
	R2	0.8675	0.7937	0.8106	0.0516	0.8075	0.8430	0.0427	0.0572	0.8452	0.8541
	Var	0.8682	0.7992	0.8178	0.0529	0.8082	0.8462	0.0429	0.0573	0.8453	0.8553

30min	MAE	3.0812	3.8443	3.9975	9.1325	8.0812	3.7666	11.9750	14.0136	3.8089	3.3411
	RMSE	5.8436	6.9098	6.9040	13.7807	11.8436	6.2646	16.4364	17.6247	6.2148	6.1151
	ACC	0.9004	0.8822	0.8823	0.7651	0.7004	0.8932	0.5946	0.5062	0.8941	0.8958
	R2	0.8205	0.7546	0.7528	0.0492	0.7205	0.7951	0.0375	0.0489	0.7979	0.8037
	Var	0.8220	0.7601	0.7595	0.0505	0.7220	0.7978	0.0374	0.0490	0.7980	0.8050
45min	MAE	3.2735	4.0607	4.2960	9.1320	8.2735	4.1217	12.5917	14.9371	4.1324	3.6399
	RMSE	6.3597	7.4254	7.5872	13.7751	11.3597	6.8776	16.7614	18.4267	6.7719	6.7389
	ACC	0.8916	0.8735	0. 8707	0.7652	0.6916	0.8828	0.5348	0.4371	0.8846	0.8852
	R2	0.7869	0.7143	0.6984	0.0478	0.7869	0.7509	0.0325	0.0427	0.7582	0.7597
	Var	0.7894	0.7196	0.7047	0.0490	0.7894	0.7538	0.0325	0.0429	0.7584	0.7612
60min	MAE	3.4260	4.2945	4.5812	9.1300	8.4260	4.4270	14.3057	15.7381	4.1324	3.9130
	RMSE	6.7397	7.9381	8.2088	13.7699	11.7397	7.3856	18.1420	18.9570	6.7719	7.2682
	ACC	0.8852	0.8648	0.8602	0.7654	0.7852	0.8742	0.4733	0.3502	0.8846	0.8762
	R2	0.7605	0.6702	0.6434	0.0462	0.6605	0.7109	0.0274	0.0391	0.7582	0.7185
	Var	0.7658	0.6753	0.6491	0.0473	0.6637	0.7137	0.0274	0.0394	0.7584	0.7202
T	Datasets	SZ-taxi									
	Metric	DASGCRN	*w/o* I_N	*w/o* A_{par}	*w/o* DA_t	*w/o* FCL	*w/o* T^D	*w/o* DA_tI_N	*w/o* DA_tA_{par}	*w/o* Mix	*w/o* TS
15min	MAE	2.4964	2.5311	2.7136	7.8361	6.5102	3.5257	9.6270	12.1762	2.5381	2.6109
	RMSE	3.9716	3.9888	4.1893	9.9781	8.0270	4.0046	11.3872	13.7591	4.0457	4.1046
	ACC	0.7233	0.7221	0.7040	0.3049	0.6143	0.7210	0.2371	0.2169	0.7182	0.7141
	R2	0.8554	0.8541	0.6733	0.0873	0.6588	0.8530	0.0674	0.0593	0.8498	0.8454
	Var	0.8559	0.8541	0.6733	0.0873	0.6592	0.8537	0.0680	0.0595	0.8500	0.8454
30min	MAE	2.5064	2.6485	2.7365	7.8344	6.3132	3.5600	10.3725	12.8530	2.5531	2.6485
	RMSE	3.9899	4.1580	4.3476	9.9757	8.6667	4.0576	11.3774	14.6328	4.0684	4.1580
	ACC	0.7221	0.7103	0.6714	0.3050	0.6034	0.7173	0.2046	0.1928	0.7166	0.7103
	R2	0.8541	0.8414	0.6573	0.0878	0.6521	0.8491	0.0592	0.0578	0.8482	0.8414
	Var	0.8546	0.8414	0.6573	0.0878	0.6524	0.8498	0.0592	0.0579	0.8484	0.8414
45min	MAE	2.5157	2.6530	2.7942	7.8392	6.1271	3.5883	11.9260	13.6430	2.5858	2.6530
	RMSE	4.0034	4.1669	4.6217	9.9800	8.0709	4.0908	12.8517	15.0647	4.1209	4.1669
	ACC	0.7211	0.7098	0.6350	0.3049	0.5965	0.7151	0.1794	0.1744	0.7130	0.7098
	R2	0.8531	0.8408	0.6114	0.0874	0.6067	0.8467	0.0516	0.0538	0.8444	0.8408
	Var	0.8536	0.8408	0.6112	0.0874	0.6070	0.8473	0.0510	0.0542	0.8445	0.8408
60min	MAE	2.5254	2.6635	2.8272	7.8342	6.6996	3.6056	11.7428	13.6430	2.5981	2.6635
	RMSE	4.0167	4.1821	4.9261	9.9778	6.3843	4.1166	14.2693	15.0647	4.1365	4.1821
	ACC	0.7202	0.7085	0.5073	0.3050	0.5512	0.7133	0.1274	0.1744	0.7119	0.7087
	R2	0.8521	0.8397	0.5800	0.0882	0.5856	0.8448	0.0375	0.0538	0.8432	0.8397
	Var	0.8526	0.8397	0.5800	0.0882	0.5860	0.8453	0.0374	0.0542	0.8435	0.8397

Table III presents the ablation study results for the DASGCRN variants on the Los-loop and SZ-taxi datasets, indicating that each design module performs as expected. To visually compare the performance of DASGCRN and its variants. Fig. 6 further illustrates the comparative results for both datasets. From the results, we can observe:

1) When the dynamic adjacency matrix DA_t is removed, the model's predictive performance declines sharply, demonstrating the necessity of capturing dynamic information within the road network.

2) The learnable parameter matrix A_{par} can adaptively explore potential spatial relationships between nodes, contributing to improved predictive performance.

3) The removal of temporal embeddings T^D results in a decrease in overall model performance. This indicates that temporal embeddings have a significant impact on the predictive capability of DASGCRN.

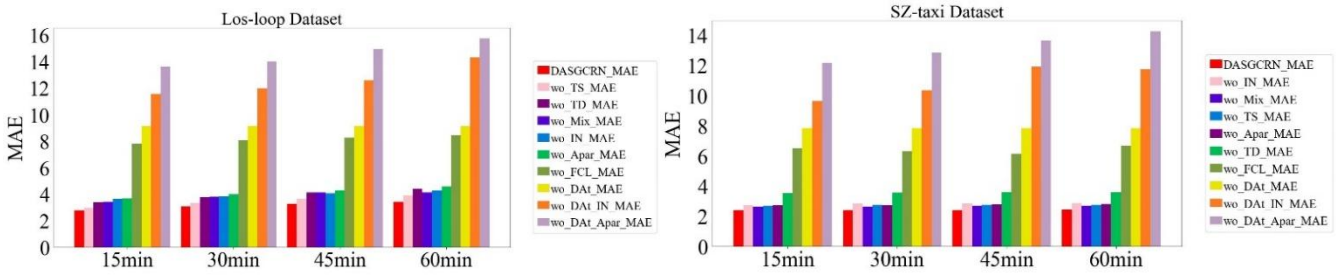

Fig. 6. Ablation experimental results on Los-loop and SZ-taxi

VI. Conclusion

A traffic speed prediction model DASGCRN based on dynamic spatial-temporal information is proposed. Firstly, the parameterized matrix and self-recurrent adjacency matrix are used to represent the spatial relationship between nodes effectively, which surpasses the traditional static adjacency matrix. Secondly, the dynamic graph is generated by using spatial-temporal feature embedding and traffic speed sequence, and the effective combination of dynamic graph and predefined graph is realized by step generation process. This method reduces the dependence on the prior knowledge of the road network and enhances the capture of the dynamic spatial-temporal dependence in the traffic network. Finally, the residual decomposition with skip connection is used to facilitate the transfer of feature information. It improves the training process of the model, and further improve the performance of the model.

In the future, we will further integrate more real-time datasets associated with traffic flow prediction, such as Internet of Things devices, autonomous vehicles, and mobile sensors, the model can capture the dynamic variations of the traffic network more accurately. Meanwhile, we will also enhance our attention to external factors such as weather and road conditions, and incorporate them as key features into the model, thereby enhancing the model's expressive capacity and predictive accuracy in a data-driven manner.

Acknowledgment

All the data mentioned in this article can be available at https://github.com/pritamBikram/Traffic_Dataset.

We would like to thank the research team that provided the dataset.

Funding

This work was supported by the Outstanding Youth Innovation Teams in Higher Education of Shandong Province(2019KJN048).

References

[1] Yin X, Wu G, Wei J, et al. Deep learning on traffic prediction: Methods, analysis, and future directions[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(6): 4927-4943.

[2] Liu J, Guan W. A summary of traffic flow forecasting methods[J]. Journal of highway and transportation research and development, 2004, 21(3): 82-85.

[3] Kamarianakis Y, Prastacos P. Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches[J]. Transportation Research Record, 2003, 1857(1): 74-84.

[4] Montesinos López O A, Montesinos López A, Crossa J. Support vector machines and support vector regression[M]//Multivariate Statistical Machine Learning Methods for Genomic Prediction. Cham: Springer International Publishing, 2022: 337-378.

[5] Yao R, Zhang W, Zhang L. Hybrid methods for short-term traffic flow prediction based on ARIMA-GARCH model and wavelet neural network[J]. Journal of Transportation Engineering, Part A: Systems, 2020, 146(8): 04020086.

[6] MENG Chuang, WANG Hui, LIN Hao, LI Kecen, WANG Xinpeng. Review of Research on Road Traffic Flow Data Prediction Methods[J]. Computer Engineering and Applications, 2023, 59(14): 51-61.

[7] CUI Jian-xun, YAO Jia, ZHAO Bo-yuan. Review on short-term traffic flow prediction methods based on deep learning[J]. Journal of Traffic and Transportation Engineering, 2024, 24(2): 50-64.

[8] Yang D, Li S, Peng Z, et al. MF-CNN: traffic flow prediction using convolutional neural network and multi-features fusion[J]. IEICE TRANSACTIONS on Information and Systems, 2019, 102(8): 1526-1536.

[9] Lu S, Zhang Q, Chen G, et al. A combined method for short-term traffic flow prediction based on recurrent neural network[J]. Alexandria Engineering Journal, 2021, 60(1): 87-94.

[10] Belhadi A, Djenouri Y, Djenouri D, et al. A recurrent neural network for urban long-term traffic flow forecasting[J]. Applied Intelligence, 2020, 50: 3252-3265.

[11] Zhao L, Song Y, Zhang C, et al. T-GCN: A temporal graph convolutional network for traffic prediction[J]. IEEE transactions on intelligent transportation systems, 2019, 21(9): 3848-3858.

[12] Zhu J, Wang Q, Tao C, et al. AST-GCN: Attribute-augmented spatial-temporal graph convolutional network for traffic forecasting[J]. IEEE Access, 2021, 9: 35973-35983.

[13] YANG Ping, LI Chengxin, LIU Yicheng, et al. A spatio-temporal graph model for multi-step prediction of dynamic traffic flow[J]. Computer Engineering and Design, 2024,45(04): 1195-1201.

[14] Wu Z, Pan S, Long G, et al. Graph WaveNet for Deep Spatial-Temporal Graph Modeling[C]//proceedings of the International Joint Conference on Artificial Intelligence. 2019: 1907-1913.

[15] Choi J, Choi H, Hwang J, et al. Graph neural controlled differential equations for traffic forecasting[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2022, 36(6): 6367-6374.

[16] Fang Z, Long Q, Song G, et al. Spatial-temporal graph ode networks for traffic flow forecasting[C]//Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021: 364-373.

[17] Zhang X, Huang C, Xu Y, et al. Traffic flow forecasting with spatial-temporal graph diffusion network[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(17): 15008-15015.

[18] Lan S, Ma Y, Huang W, et al. DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting[C]//International conference on machine learning. PMLR, 2022: 11906-11917.

[19] Li F, Feng J, Yan H, et al. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution[J]. ACM Transactions on Knowledge Discovery from Data, 2023, 17(1): 1-21.

[20] Shi Z, Zhang Y, Wang J, et al. DAGCRN: Graph convolutional recurrent network for traffic forecasting with dynamic adjacency matrix[J]. Expert Systems with Applications, 2023, 227: 120259.

[21] Zhang X, Wen S, Yan L, et al. A hybrid-convolution spatial–temporal recurrent network for traffic flow prediction[J]. The Computer Journal, 2024, 67(1): 236-252.

[22] Xia Z, Zhang Y, Yang J, et al. Dynamic spatial–temporal graph convolutional recurrent networks for traffic flow forecasting[J]. Expert Systems with Applications, 2024, 240: 122381.

[23] Fang S, Zhang Q, Meng G, et al. GSTNet: Global spatial-temporal network for traffic flow prediction[C]//IJCAI. 2019: 2286-2293.

[24] Wang X, Ma Y, Wang Y, et al. Traffic flow prediction via spatial temporal graph neural network[C]//Proceedings of the web conference 2020. 2020: 1082-1092.

[25] Bai J, Zhu J, Song Y, et al. A3t-gcn: Attention temporal graph convolutional network for traffic forecasting[J]. ISPRS International Journal of Geo-Information, 2021, 10(7): 485.

[26] Tian X, Zou C, Zhang Y, et al. NA-DGRU: A Dual-GRU Traffic Speed Prediction Model Based on Neighborhood Aggregation and Attention Mechanism[J]. Sustainability, 2023, 15(4): 2927.

[27] Wang C H, Cai J, Ye Q, et al. A two-stage convolution network algorithm for predicting traffic speed based on multi-feature attention mechanisms[J]. Journal of Intelligent & Fuzzy Systems, 2023 (Preprint): 1-16.

[28] Sun L, Liu M, Liu G, et al. FD-TGCN: Fast and dynamic temporal graph convolution network for traffic flow prediction[J]. Information Fusion, 2024, 106: 102291.