2025 6th International Conference on Pattern Recognition and Machine Learning

Attention Graph Neural Controlled Differential Equations Approach for Traffic Flow Forecasting

Chuanyu Yao
School of Computer and Information
Anhui Normal University
WuHu, China
2037231374@qq.com

Wen Zhou*
School of Computer and Information
Anhui Normal University
WuHu, China
w.zhou@ahnu.edu.cn
*Corresponding author

2025 IEEE 6th International Conference on Pattern Recognition and Machine Learning (PRML) | 979-8-3503-9258-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/PRML66062.2025.11159776

Abstract*—With the development of intelligent transportation systems, traffic flow prediction on complex traffic networks has become increasingly important, particularly in urban management and traffic planning, where it holds significant practical value. Existing traffic flow prediction methods primarily rely on stacking discrete modules to enhance prediction accuracy. However, the discontinuous hidden state trajectories between discrete modules may result in increased numerical errors and insufficient precision in capturing dynamic changes. Neural controlled differential equations (NCDEs), which can continuously adjust hidden state trajectories based on control signals, effectively address these issues. For this, we propose an Attention Graph Neural Controlled Differential Equations model (AG-NCDE). Specifically, the model first employs an attention controlled differential equation to model continuous-time dependencies. Then, it integrates high-dimensional spatiotemporal features through continuous dynamic spatiotemporal embeddings to construct a dynamic spatiotemporal graph. Finally, the spatial features are propagated using a graph controlled differential equation. The proposed AG-NCDE model is evaluated through comparative experiments on four public datasets: PEMS03, PEMS04, PEMS07, and PEMS08. The results demonstrate that the model exceeds most baseline models across three different metrics in terms of overall prediction performance. Additionally, it exhibits superior results in handling missing data.

Keywords*—*traffic flow forecasting, spatio-temporal embedding, neural controlled differential equations, graph convolution network

I. Introduction

With the acceleration of urbanization and the rapid growth of automobile ownership, urban traffic problems have become increasingly prominent. Traffic flow prediction enables intelligent transportation systems to anticipate upcoming traffic conditions, allowing for real-time adjustments to traffic signals, electronic road signs, and other control measures. This optimizes road network efficiency, reduces unnecessary stops and wait times, and improves traffic flow, thereby addressing urban traffic issues. Furthermore, analyzing and predicting traffic flow data helps urban planners understand traffic demands, optimize road designs, allocate resources effectively, and plan future transportation infrastructure construction[1]. Therefore, accurate traffic flow prediction is of profound significance to urban transportation systems.

In recent years, the success of deep learning models has been particularly notable, largely due to their ability to capture the inherent spatiotemporal dependencies in transportation systems. Among them, models based on graph convolutional networks (GCNs) have been widely applied due to their superior performance[2],[3], while methods utilizing neural controlled differential equations (NCDEs) have also gained attention [4],[5]. Researchers have dedicated significant efforts to developing novel models for traffic prediction, incorporating innovations such as advanced graph convolution techniques, graph structure learning, and efficient attention mechanisms. However, despite these advances, two key challenges remain largely unaddressed:

Modeling long-term dependencies in neural controlled differential equations: The hidden states in NCDEs are driven by the pointwise changes of control signals, primarily reflecting short-term dynamic features[6]. Over time, the impact of earlier inputs tends to be overshadowed by more recent signals, leading to a weakening of long-term dependency information. Long-term dependencies are essential for modeling complex spatiotemporal sequences, particularly in traffic flow prediction, where accurately capturing such information can significantly improve prediction performance[7].

Leveraging the temporal and spatial characteristics of traffic signals in NCDEs: Traffic flow spatiotemporal data exhibit pronounced high-dimensional patterns: temporal data reveal daily, weekly, and monthly periodic patterns, while spatial data involve dynamic node relationships and complex interactions between regions[8]. Existing NCDE-based traffic flow prediction models primarily focus on capturing changes in continuous time series but pay insufficient attention to temporal periodicity and the dynamic evolution of spatial relationships, thereby limiting their performance in traffic flow prediction.

To address these challenges, this paper proposes an Attention-based Graph Convolutional Neural Controlled Differential Equation (AG-NCDE) model for traffic flow prediction. First, AG-NCDE employs an attention-controlled differential equation to store all historical hidden states in external memory units, ensuring the model can capture global dependencies. Then, it integrates temporal information across multiple scales and dynamic spatial relationships using continuous dynamic spatiotemporal embeddings. Finally, a graph convolutional controlled differential equation propagates traffic flow state information spatially among nodes, capturing complex interactions between regions.

The main contributions of this paper are summarized as follows:

- A new neural controlled differential equation is designed to learn temporal dependencies. By integrating an external attention mechanism with NCDEs, the model overcomes the limitation of NCDEs in capturing long-range spatiotemporal dependencies.
- A novel graph structure is developed. It generates continuous dynamic spatiotemporal embeddings by blending temporal features at different scales with dynamic spatial features and constructs a dynamic graph structure based on node similarity to extract spatial features.
- Experiments on four real-world datasets demonstrate the effectiveness of the proposed model. Additionally, further tests validate the robustness of the model under missing data conditions.

II. RELATED WORK

A. Traffic forecasting

In recent years, traffic flow prediction has emerged as a key task within intelligent transportation systems, attracting significant attention from researchers[9],[10]. Traditional statistical methods, such as the Autoregressive Integrated Moving Average (ARIMA) model[11] and Support Vector Machines (SVM) [12], were among the earliest approaches used to predict traffic metrics. However, these methods fail to effectively address the complex spatiotemporal correlations in traffic flow data due to their disregard for spatial characteristics. With the rise of deep learning techniques, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been widely adopted for traffic flow prediction. For instance, ST-ResNet[13] leverages deep residual CNNs to predict urban crowd flows, while DCRNN[14] combines diffusion processes to simulate road traffic conditions and employs Gated Recurrent Units (GRUs) to capture temporal features. Nevertheless, these models demonstrate limited capability in modeling spatiotemporal relationships in non-Euclidean spaces. Recently, Graph Neural Network (GNN)-based methods have rapidly advanced, becoming the mainstream approach in traffic flow prediction. STGC[15] was the first to introduce convolutional structures and graph convolutional networks to simultaneously model temporal and spatial features. Graph WaveNet[16] extracts complex spatiotemporal correlations using adaptive graph structures and gated convolutional networks. Additionally, ASTGCN[17] and AGCRN[18] enhance traffic flow modeling capabilities by incorporating attention mechanisms and adaptive graph structures. More recently, STFGNN[19] and DSTGCN[20] have explored dynamic associations between nodes through dynamic graph generation methods, while STGODE[21] integrates neural differential equations to capture nonlinear spatiotemporal characteristics. These approaches have made remarkable progress in capturing complex spatiotemporal dependencies. However, further research is required to address challenges such as capturing dynamic changes in traffic flow data and improving the efficiency of graph structure construction.

B. Neural Controlled Differential Equations

Neural Differential Equations (NDEs) represent an approach that combines continuous-time differential equations with neural networks[22]. Neural Ordinary Differential Equations (NODEs) were the first to apply residual networks to continuous simulations of neural differential equations (ODEs) and developed a general framework for constructing such equations[23]. STGODE extended this concept to a continuous version of Graph Neural Networks (GNNs) by replacing residual connections between convolutional layers with neural differential equations, effectively capturing spatial dependencies across multiple time series. Despite the significant success of NODEs in extending time series models, the hidden states of time series can only be modeled locally and continuously because NODEs take observations from a single time step as input. Inspired by rough path theory, Neural Controlled Differential Equations introduced a generalized modeling framework for neural differential equations by interpolating discrete observations into continuous control path equations. This approach enables the use of observations from all time steps as inputs[24]. Furthermore, researchers have proposed leveraging rough path theory to extend the construction of NCDEs, addressing long time series and providing new insights into forward propagation computations for these models. NCDEs have become a powerful tool for modeling irregular time series due to their advantages in memory efficiency and their ability to handle partially observed data.

In recent years, Neural Controlled Differential Equations have achieved significant progress in traffic prediction literature. STG-NCDE was the first to combine Graph Neural Networks with controlled differential equations, overcoming the previous limitation of NCDEs in capturing long-range spatiotemporal dependencies. To address this issue, STCED proposed a holistic spatiotemporal continuous encoder-decoder based on NCDEs. However, these networks have not fully exploited the unique characteristics of traffic flow data and still rely solely on adaptive adjacency matrices.

III. PRELIMINARIES

This section introduces the spatial and temporal characteristics of traffic networks and signals, as well as the prediction problems to be addressed. It also provides an overview of the basic concepts of neural controlled differential equations.

Definition 1. **Traffic Network**. A traffic network can be modeled as a directed weighted graph $G = (V, E, A)$, where V and E represent the set of nodes and edges, respectively, and $|V| = N$ is the number of nodes. $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix of the graph, describing the relationships between nodes, typically determined by connectivity or distance to define edge weights.

Definition 2. **Temporal and Spatial Properties of Traffic Signals.** Assuming there are N nodes in the traffic network, and the sensors sample data N_d times per day over seven days of the week, the spatial and temporal characteristics can be stored in three independent trainable embedding matrices: $E_a \in \mathbb{R}^{N\times D}, E_d \in \mathbb{R}^{N\times D}, E_w \in \mathbb{R}^{N\times D}$, where D represents the embedding dimension.

Definition 3. **Traffic Flow Prediction Problem.** Given the traffic flow sequence $X_{t-T+1:t}$ from the previous T time steps, the goal of traffic flow prediction is to infer the traffic flow for the next T' time steps by training a parameterized model f with parameters θ. This can be expressed as:

$$[X_{t+1},\dots,X_{t+T'}] = f([X_{t-T+1},\dots,X_t],\theta,G) \quad (1)$$

where each time step $X_i \in \mathbb{R}^{N\times F}$, N represents the number of spatial nodes, and F is the dimension of input features.

Definition 4. **Neural Controlled Differential Equations**. Neural controlled differential equations describe the evolution of the hidden state z_t driven by the control signal X_s, which can be expressed as:

$$z_t = z_{t_0} + \int_{t_0}^{t} f_\theta(z_s)\, dX_s, \quad t \in (t_0, t_n] \quad (2)$$

where z_t represents the system's state at time t, and z_{t_0} is the initial state of the system at time t_0. The dynamic evolution is determined by the vector field $f_\theta(z_s)$, where f_θ is a function parameterized by θ, typically learned by a neural network, describing how the system state changes with the control signal. The changes in the control signal are captured by the differential term dX_s, where X_s is the input time series, interpolated to construct a continuous trajectory. The time interval $t \in (t_0, t_n]$ represents the evolution process from the initial timet_0 to the end time t_n.

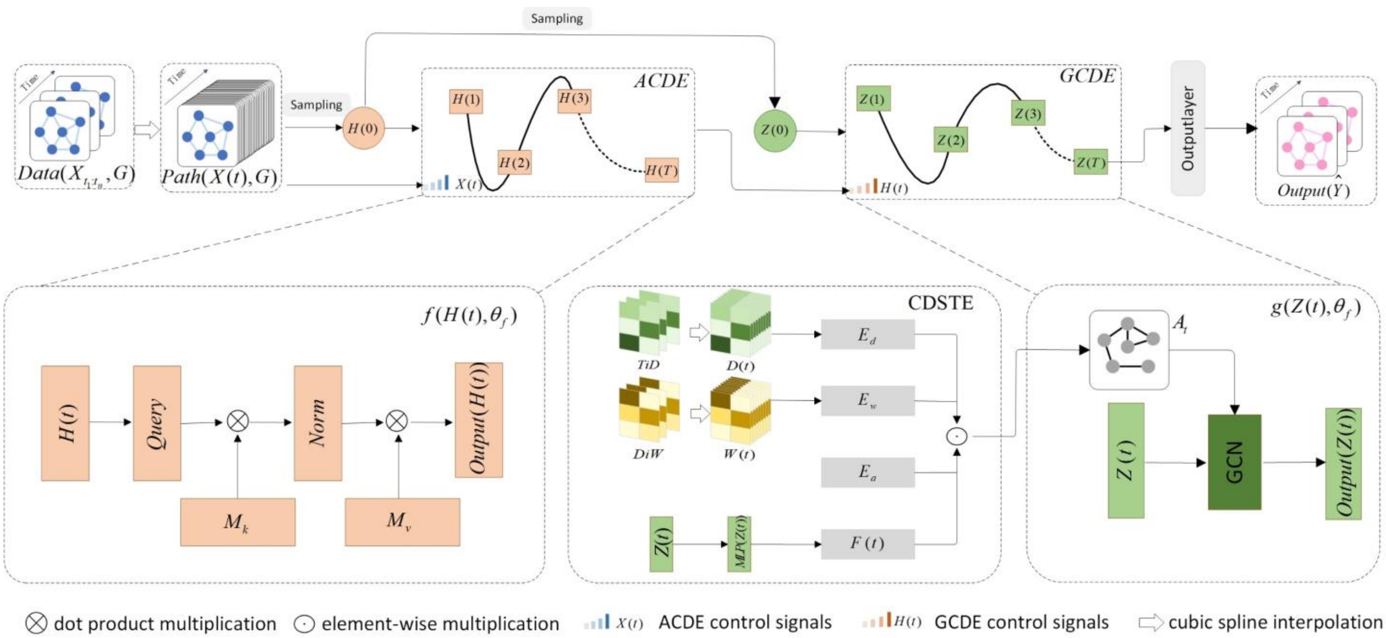


Fig. 1. Framework of AG-NCDE model. $\boldsymbol{X(t)}$ represents the continuous time signal generated by cubic spline interpolation of the input data. $\boldsymbol{H(t)}$ denotes the state of ACDE at time t, which abstracts the dynamic features of the input driving signal $X(t)$. $\boldsymbol{f(H(t),\theta_f)}$ is the control vector field of ACDE, defining the evolution rules of $H(t)$. **ACDE** specifies how the current state $H(t)$ interacts with the variations of the driving signal $X(t)$ through the control vector field, determining the evolution path of the state. Similarly, $\boldsymbol{Z(t)}$ represents the state of GCDE at time t, with $H(t)$ serving as the input control signal for GCDE. $\boldsymbol{g(Z(t),\theta_g)}$ is the control vector field of GCDE. $\boldsymbol{TiD}$ and $\boldsymbol{DiW}$ represent daily and weekly periodic information, respectively. $\boldsymbol{E_d}$, $\boldsymbol{E_w}$ and $\boldsymbol{E_a}$ represent the daily embedding, weekly embedding, and spatial embedding matrices, respectively. **CDSTE** refers to Continuous Dynamic Spatio-Temporal Embedding, whose output generates a dynamic graph A_t based on node similarity.

IV. PROPOSED FRAMEWORK

The overall framework of AG-NCDE is depicted in Fig. 1. Detailed explanations of components are presented in the subsequent subsections.

A. Attention Controlled Differential Equation

The attention controlled differential equation first constructs a continuous time path $X(t)$ using the natural cubic spline interpolation algorithm. This path is then input into the attention controlled differential equation as a control signal. This approach enhances the external attention mechanism[25], allowing it to model dependencies in time-series data within the framework of neural controlled differential equations.

The specific formulation of the attention controlled differential equation is expressed as follows:

$$H(T) = H(0) + \int_0^T f(H(t),\theta_f)dX(t) \quad (3)$$

where $H(t) \in \mathbb{R}^{N\times D_h}$ represents the temporal modeling output of the nodes at time $t \in [0,T]$, where D_h denotes the hidden layer dimension. The initial state $H(0)$ is derived from $X(0)$ through a fully connected layer, given by: $H(0) = FC_h(X(0))$. The external attention mechanism is incorporated into the controlled differential equation function $f(H(t),\theta_f)$, defined as:

$$f(H(t),\theta_f) = \sigma(H(t)M_k^T)M_V \quad (4)$$

where $M_k, M_v \in \mathbb{R}^{N \times F}$ are learnable parameters representing the keys and values of the external attention mechanism, respectively. These parameters serve as memory units for the entire training dataset. The attention scores $A_{tt} = H(t)M_k^T$ are derived from the learned data priors and normalized using a function similar to self-attention, specifically $\sigma(\mathrm{x}) = \mathrm{sigmoid}(\mathrm{x})$. The normalized attention scores A_{tt} are then used to update the input features from M_v.

B. Continuous Dynamic Spatiotemporal Embedding

AG-NCDE includes a continuous dynamic graph generation method to capture dynamic spatial dependencies in sensor data. This approach removes the reliance on predefined adjacency matrices, making it suitable for scenarios where prior knowledge is unavailable. The specific implementation is as follows:

For traffic flow input, the continuous dynamic spatiotemporal embedding adds corresponding daily periodic information $TiD \in \mathbb{R}^{N \times T \times N_d}$ and weekly periodic information $DiW \in \mathbb{R}^{N \times T \times N_w}$ to each timestamp. where $N_d = 288$ represents the number of sampling points in a day, and $N_w = 7$ represents the days of the week. Using the cubic spline interpolation method, continuous time paths for these periodic features are generated as $D(t) \in \mathbb{R}^{N \times N_d}, W(t) \in \mathbb{R}^{N \times N_w}, t \in [0, T]$. Subsequently, daily embeddings $P_d \in \mathbb{R}^{N_d \times D}$ and weekly embeddings $P_w \in \mathbb{R}^{N_w \times D}$ at time t are derived via the daily embedding layer $E_d \in \mathbb{R}^{N \times D}$ and the weekly embedding layer $E_w \in \mathbb{R}^{N \times D}$, where D denotes the embedding dimension.

To better reflect these local variations in traffic flow prediction, the continuous dynamic spatiotemporal graph replaces the original fixed spatial distribution graph with adaptive embeddings $E_a \in \mathbb{R}^{N \times D}$, where D denotes the embedding dimension. Additionally, to ensure the graph's dynamic nature, the traffic state at time t is processed through a fully connected layer (MLP) to extract dynamic signals: $F(t) = MLP(Z(t))$, where $F(t) \in \mathbb{R}^{N \times D}$.

Finally, these embeddings and dynamic signals are combined elementwise to produce the continuous dynamic spatiotemporal embedding $E_{dhste} \in R^{N \times D}$, calculated as:

$$E_{dhste} = E_d \odot E_w \odot E_a \odot F(t) \tag{5}$$

Once the continuous dynamic spatiotemporal embedding is obtained, a graph can be generated based on node similarity. Dynamic spatiotemporal embeddings adapt in real time to the evolving patterns of traffic networks. By dynamically updating the adjacency matrix and embedding representations, the model integrates the periodic characteristics of traffic flow, enabling a more precise representation of spatiotemporal dependencies. The continuous time paths generated using cubic spline interpolation align seamlessly with neural controlled differential equations, which are designed to effectively process continuous-time signals.

C. Graph Controlled Differential Equation

Graph Controlled Differential Equations combines graph convolutional networks with neural controlled differential equations by integrating the spatial feature extraction capabilities of graph convolution into the controlled differential equation framework. This integration forms a model architecture capable of simultaneously capturing spatiotemporal dynamics. The model not only effectively models spatial dependencies in graph-structured data but also leverages the continuous-time dynamic modeling capabilities of neural controlled differential equations, enabling it to capture underlying trends and nonlinear characteristics in time series data.

The mathematical formulation of GCDE is as follows:

$$Z(T) = Z(0) + \int_0^T g\big(Z(t), \theta_g\big) dH(t) \tag{6}$$

where $Z(t) \in \mathbb{R}^{N \times D_z}$ denotes the output of nodes processed by the neural controlled differential equation and graph convolution. Here, D_z represents the hidden layer dimension, and the initial state $Z(0)$ is derived from $H(0)$ via a fully connected layer. The key element in this formulation is the controlled differential equation (CDE) function $g\big(Z(t), \theta_g\big)$, parameterized by θ_g to perform spatial processing. The specific design of the CDE function with graph convolution is:

$$g\big(Z(t), \theta_g\big) = \left(I_N + D_t^{-\frac{1}{2}} A_t D_t^{-\frac{1}{2}}\right) Z(t) W_g + \mathrm{b_g} \tag{7}$$

where $A_t \in \mathbb{R}^{N \times N}$ represents the dynamic graph at time step t. This graph is inferred from spatial dependencies via the continuous dynamic spatiotemporal embedding. Specifically, it is derived using cosine similarity and computed as $A_t = ReLU(E_{dhste} E_{dhste}^T)$, where $E_{dhste} E_{dhste}^T$ represents the continuous dynamic spatiotemporal embedding. The degree matrix D_t is used to compute the symmetric normalized dynamic adjacency matrix $D_t^{-\frac{1}{2}} A_t D_t^{-\frac{1}{2}}$, which preserves the structural information and enhances training stability through normalization. $I_N \in R^{N \times N}$ is the identity matrix, which, when introduced after normalization, further improves numerical stability. Finally, W_g and b_g denote the weight matrix and bias vector, respectively, within the graph convolution operation.

D. Output Layer

After processing through the Graph Convolutional Controlled Differential Equation, the final output, $\hat{Y}$, is computed from $Z(T)$ using the following output layer:

$$\hat{Y} = Z(T) W_{\mathrm{o}} + b_{\mathrm{o}} \tag{8}$$

where $W_o \in \mathbb{R}^{D_z \times l \times C}$ and $b_o \in \mathbb{R}^{l \times C}$ denote the trainable weights and biases of the output layer, respectively. The parameter l represents the prediction horizon (number of steps), while C indicates the number of prediction features, which in this case is fixed to 1 (traffic flow).

To ensure effective training of the network, the Mean Squared Error (MSE) loss function is adopted to optimize the model. This loss function is defined as follows:

$$\mathrm{Loss} = \frac{1}{M \cdot N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| y^{(i,j)} - \hat{y}^{(i,j)} \right| \tag{9}$$

In the above equation, M represents the total number of training samples, and N denotes the number of nodes in each training sample.

V. EXPERIMENTS

A. Datasets

This study evaluates the proposed method on four representative spatiotemporal datasets: PEMS03, PEMS04, PEMS07, and PEMS08. These datasets were collected by the California Performance Measurement System (PEMS) from two regions in California. The time interval for all four datasets is 5 minutes, resulting in 12 records per hour. Detailed information is provided in Table I.

TABLE I. STATISTICS FOR PMSE DATASETS

Datasets	Nodes	Time steps	Time interval
PEMS03	358	26,209	5min
PEMS04	307	16,992	5min
PEMS07	883	28,224	5min
PEMS08	170	17,856	5min

B. Experimental Settings

All experiments in this study were conducted on a Windows PC equipped with a GeForce RTX 4060 TI GPU, using the PyTorch framework. For a fair comparison with baseline models, the datasets were split into training, validation, and testing sets in a 6:2:2 ratio. In all datasets, the traffic flow for the next hour was predicted based on observations from the previous hour, meaning 12 consecutive observations were used to forecast the next 12 time steps.

Regarding model hyperparameters, the embedding dimension D was set to 10, while the hidden layer dimensions D_h for the Attention-controlled Differential Equation and D_z for the Graph Convolutional Controlled Differential Equation were both set to 128. The Chebyshev polynomial order for graph convolution was set to 2. The model was trained using the Adam optimizer for up to 150 epochs, with a batch size of 64 per epoch and a learning rate of 0.001. Early stopping was employed during training, where the process was terminated if the validation loss did not decrease for 15 consecutive epochs.

To effectively evaluate the proposed method and verify its superiority, three metrics were used: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). These metrics reflect the differences between predicted and actual values, with lower values of MAE, MAPE, and RMSE indicating better performance of the prediction model.

C. Forecasting Performance Comparison

Here, the AG-NCDE model was evaluated and compared with 10 different baseline models (STGCN, ASTGCN, AGCRN, DSTAGNN [26], ST-AE [27], LSGCN [28], STGODE, SGP [29], GT-LSTM(p) [30] and ST-GCN [31]) on four PEMS datasets using three evaluation metrics: MAE, RMSE, and MAPE. The results are shown in Table II.

It is not difficult to see that the AG-NCDE model proposed in this paper outperforms all baselines in all evaluation metrics. The main reasons for the significant advantages of the proposed method are as follows: (1) STGCN uses a pure convolutional structure to model multi-scale traffic networks, which obviously cannot effectively capture spatiotemporal correlations; (2) Spatiotemporal graph models such as ASTGCN and LSGCN are based on predefined static graphs, ignoring the spatiotemporal similarities between nodes, thus resulting in suboptimal performance; (3) The DSTAGNN model utilizes a data-driven strategy with a dynamic spatiotemporal perception graph to replace the predefined static graph. However, spatiotemporal graphs constructed based on prior knowledge may not align with the temporal patterns of traffic flow, leading to unsatisfactory performance; (4) AGCRN and STGODE use adaptive matrices to model spatial relationships, achieving performance improvements in traffic flow prediction, but there are still shortcomings. The results indicate that AG-NCDE, due to its reasonable structural design, can better integrate the spatiotemporal correlations of traffic networks. An important observation is that GT-LSTM(p), by introducing multidimensional spatiotemporal embeddings with periodic information, achieves performance second only to AG-NCDE, suggesting that incorporating continuous dynamic spatiotemporal embedding information can effectively enhance the performance of spatiotemporal models.

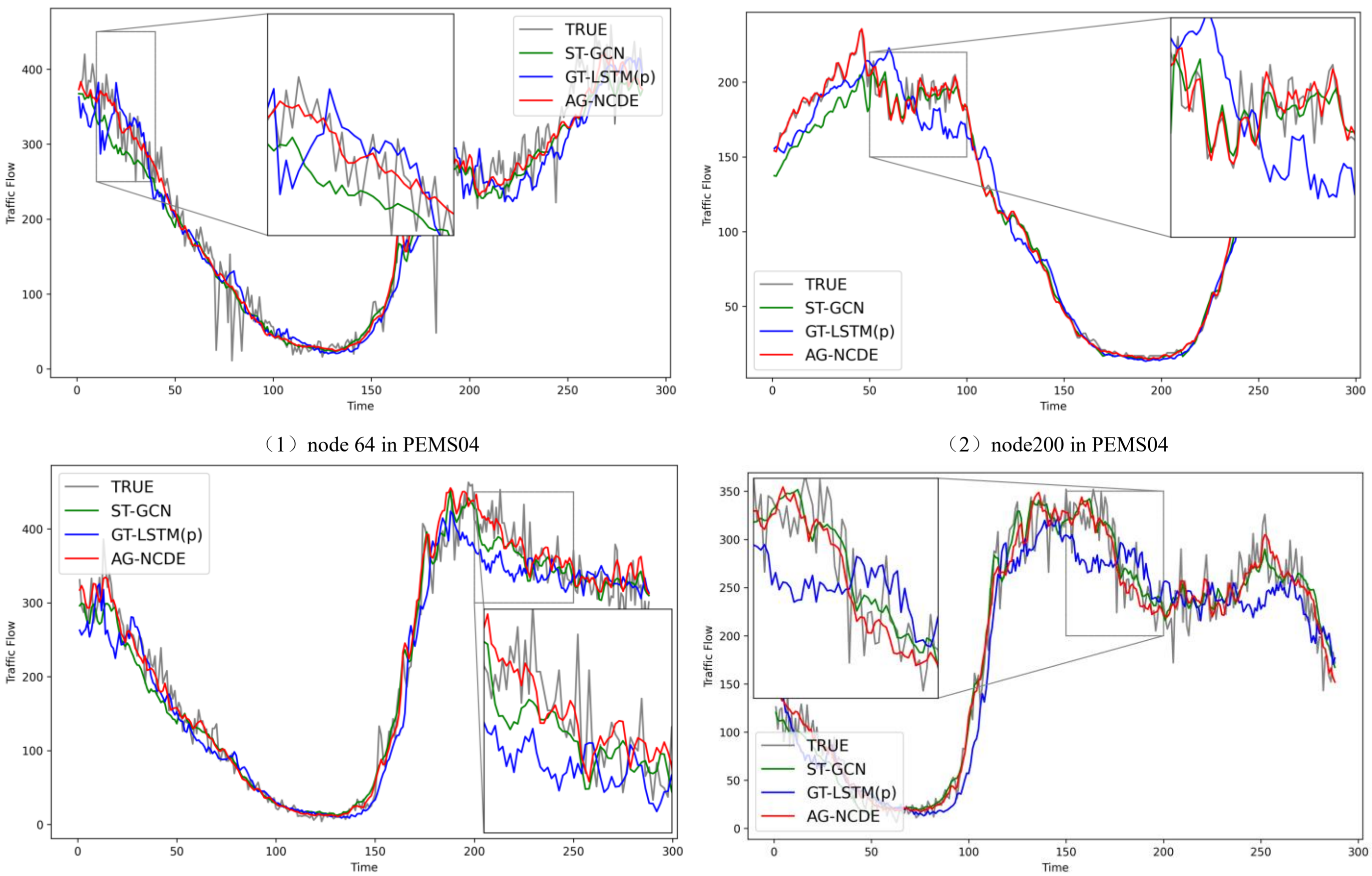
To more effectively demonstrate the superiority of the proposed method, this paper randomly sampled the traffic flow changes of nodes over a day from the PEMS04 and PEMS08 datasets. By comparing the predicted values of AG-NCDE and other baseline models with the actual values, the prediction accuracy of different models for the same traffic flow is illustrated, as shown in Figure 2. It can be observed that the proposed model demonstrates more accurate predictions for challenging cases and can adapt to the complex traffic change patterns of different nodes. This is precisely because the model used in this paper integrates multidimensional spatiotemporal information, enabling better modeling of the traffic patterns at each node.

Furthermore, to more intuitively showcase the performance of the prediction models across different prediction time steps, this paper compares the error trends of some baseline models in Figure 3. Overall, as the prediction duration increases, the slope of the performance curve also increases, indicating that the prediction accuracy of each model decreases as the time span extends. The prediction error of the proposed AG-NCDE model is lower than that of all baseline models, and the error does not significantly increase with the extension of the prediction time. This is because the model leverages the periodic changes in traffic flow to further refine the prediction results.

TABLE II COMPARISON OF PREDICTION ACCURACY OF DIFFERENT MODELS ON THE PEMS03, PEMS04, PEMS07, AND PEMS08 DATASETS.

Models	PEMS03			PEMS04			PEMS07			PEMS08		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
STGCN	15.83	27.51	16.13	19.57	31.38	13.44	21.74	35.27	9.24	16.08	25.39	10.60
ASTGCN	17.34	29.56	17.21	22.93	35.22	16.56	24.01	37.87	10.73	18.25	28.06	11.64

AGCRN	15.24	26.65	15.89	19.38	31.25	13.40	20.57	34.40	8.74	15.32	24.41	10.03
DSTAGNN	15.57	27.21	15.68	19.30	31.46	12.70	21.42	34.51	9.01	15.67	24.77	9.94
ST-AE	17.29	28.50	16.92	21.24	33.42	14.01	24.60	38.55	10.54	17.24	27.34	11.47
LSGCN	17.94	29.85	16.98	21.53	33.86	13.18	27.31	41.46	11.98	17.73	26.76	11.20
STGODE	16.50	27.84	16.69	20.84	32.82	13.77	22.59	37.54	10.14	16.81	25.97	10.62
SGP	15.85	26.92	15.74	19.57	31.52	13.19	13.13	34.97	9.92	14.96	24.03	10.27
GT-LSTM(p)	16.77	27.89	16.04	19.36	31.12	13.14	20.17	34.30	9.11	14.19	23.46	9.35
ST-GCN	16.12	26.75	15.41	19.64	31.27	13.28	21.18	34.16	9.43	14.33	23.87	9.57
AG-NCDE	**15.23**	**26.60**	**14.93**	**18.84**	**30.68**	**12.65**	**19.64**	**32.82**	**8.51**	**13.91**	**23.14**	**9.06**



（1）node 64 in PEMS04　　（2）node200 in PEMS04

（3）node 64 in PEMS04　　（4）node200 in PEMS04

Fig. 2. Traffic forecasting visualization.

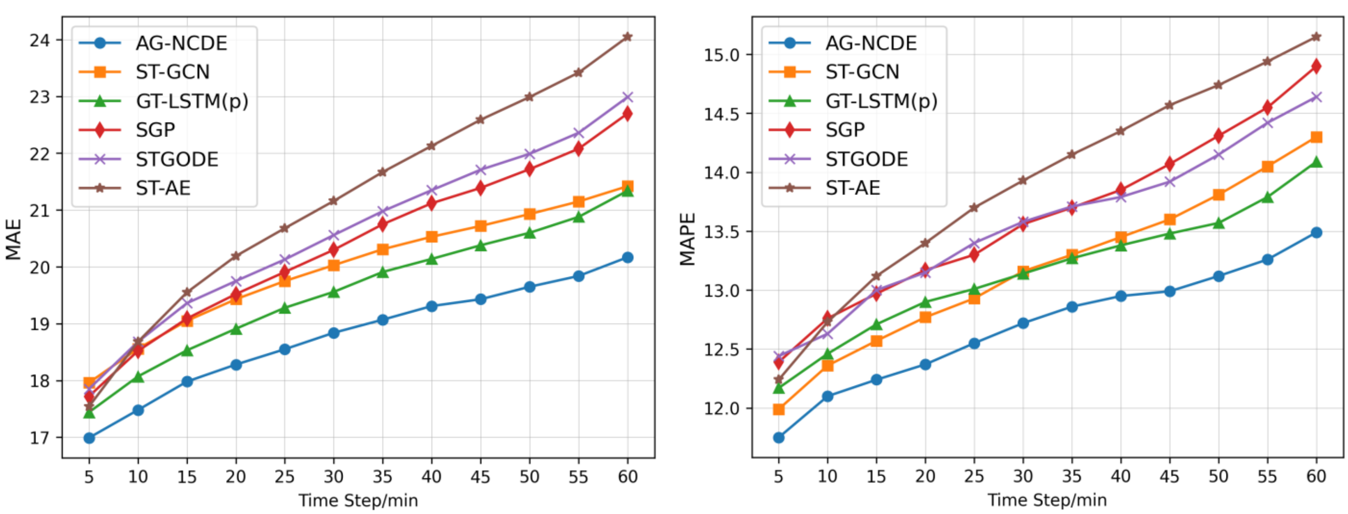


Fig. 3. Comparison of indicators of different models on the PEMS04 dataset.

D. Irregular Traffic Forecasting

In real-world data, sensor malfunctions and network anomalies can lead to significant missing observations. To demonstrate the robustness of the model, this study further evaluated the performance of AG-NCDE under different missing data percentages (30% and 50%) and compared it with the STG-NCDE model. Since AG-NCDE employs a cubic spline interpolation method to handle discrete time series and leverages neural controlled differential equations for continuous-time modeling, the model can adapt to irregular data without needing modifications to the model itself. Additionally, AG-NCDE introduces periodic factors when generating continuous dynamic spatiotemporal embeddings, which can compensate for the temporal dependency loss caused by missing data. As shown in Table III, the model proposed in this study demonstrates significantly lower errors compared to the STG-NCDE model under different missing data percentages, showing superior performance.

TABLE III. PERFORMANCE OF THE MODEL WITH DIFFERENT MISSING VALUES ON THE PEMS04 DATASET

Model	**STG-NCDE**	**AG-NCDE**	**STG-NCDE**	**AG-NCDE**
Missing rate	30%		50%	
MAE	19.36	**18.92**	19.98	**19.34**
RMSE	31.28	**30.87**	32.09	**31.04**
MAPE(%)	12.79	**12.75**	13.48	**13.26**

E. Ablation Study

This section aims to demonstrate the effectiveness of the various modules of AG-NCDE through experiments. The experiments were conducted on the PEMS03 and PEMS08 datasets, with similar results observed for other datasets. Specifically, three variants of the AG-NCDE model were designed:

(1) **w/o E_{dhste}** : The continuous dynamic spatiotemporal embedding module is removed, and only adaptive spatial embeddings are used to replace the generation of the spatiotemporal graph.
(2) **w/o GCN**: The graph convolution module is removed, and the graph controlled differential equation network is replaced with fully connected layers.
(3) **w/o TA**: The attention module is removed, and the external attention in the attention controlled differential equation is replaced with fully connected layers.

TABLE IV. RESULTS OF ABLATION EXPERIMENT

Dataset	**Model**	**MAE**	**RMSE**	**MAPE(%)**
PEMS03	w/o E_{dhste}	15.44	26.83	15.96
	w/o GCN	15.58	26.93	15.05
	w/o TA	15.34	26.87	15.13
	AG-NCDE	**15.23**	**26.60**	**14.90**
PEMS08	w/o E_{dhste}	14.33	23.64	9.48
	w/o GCN	13.27	23.67	9.46
	w/o TA	14.23	23.52	9.58
	AG-NCDE	**14.19**	**23.46**	**9.35**

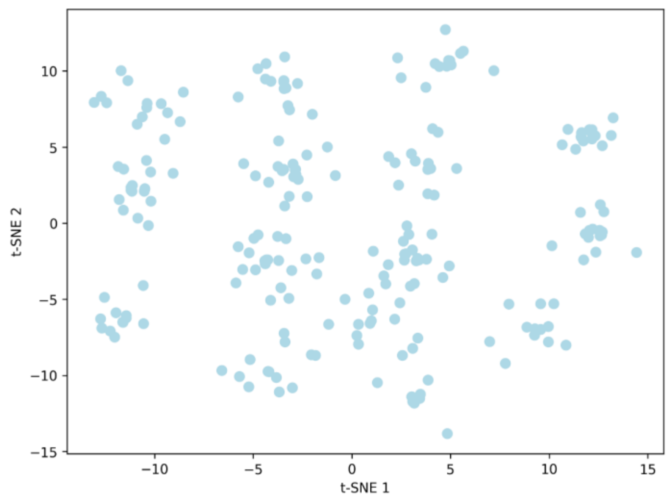


Fig. 4. Visualization of dynamic spatiotemporal embedding.

The results in Table IV lead to the following conclusions: The continuous dynamic spatiotemporal embedding, by integrating multidimensional spatiotemporal features, effectively models complex traffic flow. Removing this module leads to a decrease in model performance, highlighting its importance in improving predictive capabilities. Graph convolution aggregates the features of neighboring nodes through an information propagation mechanism, enhancing the model's ability to capture spatial dependencies. Removing the graph convolution results in a decline in predictive performance, indicating its critical role in extracting spatial features. The attention module adaptively adjusts spatiotemporal weights, capturing dynamic traffic characteristics. Its removal weakens the model's prediction ability, indicating the indispensable role of the attention mechanism in modeling spatiotemporal characteristics.

Finally, Fig. 4 shows the visualization of the proposed continuous dynamic spatiotemporal embedding, verifying its effectiveness. In this experiment, t-SNE was used to reduce the spatiotemporal embeddings to two dimensions for analysis. The embeddings of different nodes in the PEMS04 dataset naturally formed clusters. Nodes that are geographically close in the traffic network tend to exhibit similar traffic patterns, with the traffic condition of one node often influenced by that of neighboring nodes.

Clearly, our clustering results confirm these characteristics. In summary, these three modules together form the core of the model. The complete combination of modules allows for a more comprehensive capture of the multidimensional spatiotemporal features in traffic flow, significantly enhancing the model's prediction performance.

F. Parameter Analysis

In graph convolution models, the hyperparameter K typically represents the order of the convolution, i.e., the range of information propagation. The analysis of the impact of KK is shown in Table V: When $K = 1$, only the features of direct neighbors can be aggregated, leading to a lack of the model's ability to perceive a broader graph structure, resulting in underperformance. However, when $K > 2$, excessive information propagation may lead to signal over-smoothing, which weakens the expression of local structures. Additionally, larger K increases computational overhead and model complexity, which is unfavorable for practical applications. The experimental results indicate that when $K = 2$, the model achieves an optimal performance by balancing information richness and noise control.

TABLE V. SENSITIVITY TO K IN PEMS07

Dataset	K	MAE	RMSE	MAPE/%
PEMS07	1	19.83	32.90	8.66
	2	**19.64**	32.82	**8.51**
	3	19.79	**32.78**	8.84

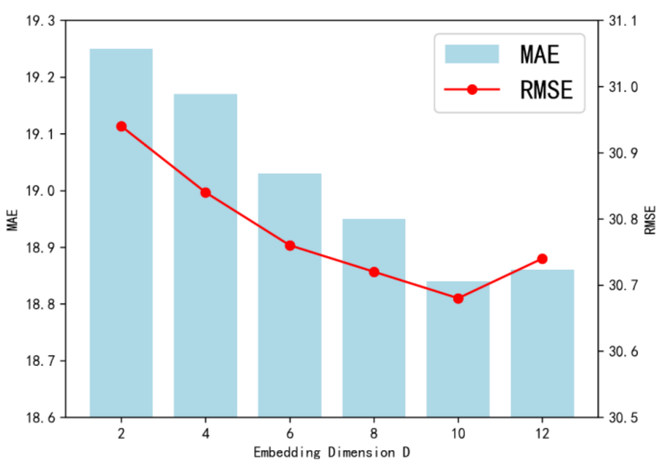


Fig. 5. Sensitivity to D in PeMS04

On the other hand, Fig. 5 shows the results for different embedding dimensions D. The model performs optimally only when the embedding dimension D is within a specific range. If the embedding dimension is too small, the model cannot effectively encode complex spatiotemporal information, limiting its feature representation capability. Conversely, if the embedding dimension is too large, it may introduce redundant features, increasing the risk of overfitting and model complexity. A reasonable embedding dimension D ensures the sufficient encoding of key information while Keeping the model efficient and robust

ACKNOWLEDGMENT

The authors appreciate the comments and suggestions of all the anonymous reviewers who helped improve this paper. In addition, this research was supported in part by the National Natural Science Foundation of China (NSFC) (grant nos. 61902003 and 61976006) and the Anhui Provincial University Key Scientific Research project of China (grant no. 2023ah050142).

REFERENCES

[1] Y. Wang et al., "Freeway traffic control in presence of capacity drop," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1497–1516, Mar. 2021.

[2] D. Peng and Y. Zhang, "MA-GCN: A memory augmented graph convolutional network for traffic prediction," *Eng. Appl. Artif. Intell.*, vol. 121, p. 106046, 2023.

[3] J. Chen, W. Wang, K. Yu, X. Hu, M. Cai, and M. Guizani, "Node Connection Strength Matrix-Based Graph Convolution Network for Traffic Flow Prediction," *IEEE Trans. Veh. Technol.*, vol. 72, no. 9, pp. 12063–12074, Sep. 2023.

[4] J. Choi, H. Choi, J. Hwang, and N. Park, "Graph Neural Controlled Differential Equations for Traffic Forecasting," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, pp. 6367–6374.

[5] J. Ji, J. Wang, Z. Jiang, J. Jiang, and H. Zhang, "STDEN: Towards Physics-Guided Neural Networks for Traffic Flow Prediction," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, pp. 4048–4056.

[6] P. Kidger, J. Morrill, J. Foster, and T. Lyons, "Neural controlled differential equations for irregular time series," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, 2020, pp. 562–573.

[7] L. Liu, Y. Shen, and H. Qi, "A Continuous Encoder-Decoder Method for Spatial-Temporal Forecasting," in *2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS)*, Nanjing, China: IEEE, Jan. 2023, pp. 835–842.

[8] W. Weng, "A Decomposition Dynamic Graph Convolutional Recurrent Network for Traffic Forecasting," *Pattern Recognit.*, vol. 142, p. 109670, Oct. 2023.

[9] D. Sun, S. Xue, H. Wu, and J. Wu, "A Data Stream Cleaning System Using Edge Intelligence for Smart City Industrial Environments," *IEEE Trans. Ind. Inform.*, vol. 18, no. 2, pp. 1165–1174, Feb. 2022.

[10] Z. Xia, S. Xue, J. Wu, Y. Chen, J. Chen, and L. Wu, "Deep Reinforcement Learning for Smart City Communication Networks," *IEEE Trans. Ind. Inform.*, vol. 17, no. 6, pp. 4188–4196, Jun. 2021.

[11] J. D. Hamilton, *Time Series Analysis*. Princeton, NJ: Princeton University Press, 2020.

[12] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto, and S. M. Easa, "Supervised Weighting-Online Learning Algorithm for Short-Term Traffic Flow Prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1700–1707, Dec. 2013.

[13] J. Zhang, Y. Zheng, and D. Qi, "Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction," *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, Feb. 2017.

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

[15] B. Yu, H. Yin, and Z. Zhu, "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Stockholm, Sweden: International Joint Conferences on Artificial Intelligence Organization, Jul. 2018, pp. 3634–3640.

[16] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*, Macao, China, 2019, pp. 1907–1913.

[17] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 922–929, Jan. 2019.

[18] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20)*, Vancouver, BC, Canada, 2020, Art. no. 1494, pp. 1–12.

[19] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," *arXiv*, vol. abs/2012.09641, 2020.

[20] J. Hu, X. Lin, and C. Wang, "DSTGCN: Dynamic spatial-temporal graph convolutional network for traffic prediction," *IEEE Sensors Journal*, vol. 22, no. 13, pp. 13116–13124, Jul. 1, 2022, doi: 10.1109/JSEN.2022.3176016.

[21] Z. Fang, Q. Long, G. Song, and K. Xie, "Spatial-temporal graph ODE networks for traffic flow forecasting," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)*, Virtual Event, Singapore, 2021, pp. 364–373.

[22] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS '18)*, Montréal, Canada, 2018, pp. 6572–6583.

[23] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2018.

[24] A. Fermanian, T. Lyons, J. Morrill, and C. Salvi, "New directions in the applications of rough path theory," *IEEE BITS the Information Theory Magazine*, vol. 3, pp. 41–53, 2023.

[25] M.-H. Guo, Z.-N. Liu, T.-J. Mu, and S.-M. Hu, "Beyond self-attention: External attention using two linear layers for visual tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5436–5447, May 2023.

[26] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, and P. Li, "DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting," in *Proc. 39th Int. Conf. Mach. Learn.* (ICML), vol. 162, 2022, pp. 11906–11917.

[27] M. Liu, T. Zhu, J. Ye, Q. Meng, L. Sun, and B. Du, "Spatio-temporal autoencoder for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5516–5526, May 2023.

[28] R. Huang, C. Huang, Y. Liu, G. Dai, and W. Kong, "LSGCN: Long short-term traffic prediction with graph convolutional networks," in *Proc. 29th Int. Joint Conf. Artif. Intell. (IJCAI)*, Yokohama, Japan, Jul. 2020, pp. 2355–2361.

[29] A. Cini, I. Marisca, F. M. Bianchi, and C. Alippi, "Scalable spatiotemporal graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 6, 2023, pp. 811–819.

[30] Y. Luo, J. Zheng, X. Wang, Y. Tao, and X. Jiang, "GT-LSTM: A spatio-temporal ensemble network for traffic flow prediction," *Neural Networks*, vol. 171, pp. 251–262, Mar. 2024.

[31] R. Zhong, B. Hu, F. Wang, Y. Feng, Z. Li, X. Song, Y. Wang, S. Lou, and J. Tan, "Multi-factor embedding GNN-based traffic flow prediction considering intersection similarity," *Neurocomputing*, vol. 620, p. 129193, 2025.