Flow-Attention based Dynamic Graph Convolutional Recurrent Network for Traffic Forecasting

Yiming Ma, Azhar Halik, and Xin Quan

Abstract—**Accurate traffic prediction is crucial for the functionality of Intelligent Transportation Systems(ITS). However, the significant nonlinearity and complexity of spatio-temporal relationships in traffic flow frequently constrain the predictive accuracy of many models, which often combine temporal and spatial components without effectively capturing spatio-temporal characteristics. To address this, we developed a Flow Attention-based Dynamic Graph Convolutional Recurrent Network (FADGCRN) model. FADGCRN combines a GRU model with flow attention and a dynamic graph convolutional recurrent network, utilizing efficient depthwise separable convolutions. It employs KAN's learnable activation functions to generate dynamic graphs from time-varying traffic signals, enabling the simultaneous extraction of both spatial and temporal features. Additionally, the model integrates a flow attention mechanism based on the principle of network flow conservation, which is used for attention weight calculation within the GRU. This design allows the model to effectively allocate attention resources across different time steps, capturing both short-term and long-term dependencies in traffic flow more efficiently. Furthermore, by using LSTM, the model can map input data into various representational spaces. This approach enhances the model's ability to capture temporal correlations when processing long sequences of data, while also minimizing the influence of specific inductive biases that traditional attention mechanisms may face. Through rigorous evaluation using four authentic traffic datasets, our model has shown remarkable performance, clearly exceeding that of other baseline models.**

Index Terms—**Traffic forecasting, Dynamic graph generation, Graph convolution network, Flow attention mechanism.**

I. Introduction

WITH the rapid urbanization and the surge in vehicle numbers, transportation systems are encountering unprecedented challenges. Traffic congestion and frequent accidents have significantly impacted travel efficiency and safety. Particularly on highways, severe traffic congestion not only disrupts the flow but also heightens the risk of accidents. Intelligent Transportation Systems (ITS) are crucial in tackling these issues. ITS enhances traffic management and optimization by integrating information technology and sensor technology. Traffic forecasting, a key component of ITS, has attracted increasing attention in recent years.

Manuscript received October 26, 2024; revised March 13, 2025.

This work was supported by the National Natural Science Foundation of China (12462028)

Yiming Ma is a postgraduate student of School of College of Mathematics and System Sciences, Xinjiang University, Urumqi 830017, China. (e-mail:3151587669@qq.com).

Azhar Halik is an Associate Professor of School of College of Mathematics and System Sciences, Xinjiang University, Urumqi 830017, China. (Corresponding author,e-mail: azhar@xju.edu.cn).

Xin Quan is a postgraduate student of School of College of Mathematics and System Sciences, Xinjiang University, Urumqi 830017, China. (e-mail: 2951446893@qq.com).

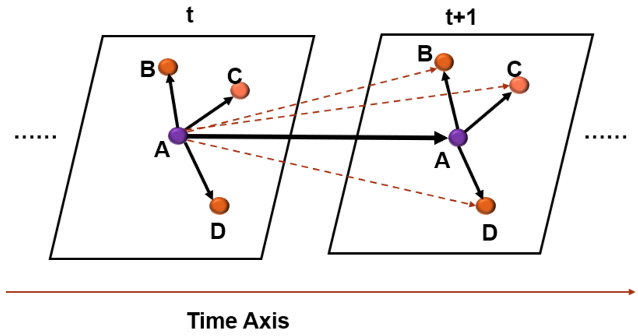


Fig. 1. Spatial–temporal correlation

Accurate prediction of traffic flow can greatly reduce congestion, thereby enhancing the overall efficiency and safety of transportation systems.

The fundamental challenge in traffic flow prediction, a spatio-temporal forecasting problem, is the accurate modeling of complex spatio-temporal correlations in traffic data. The inability of traditional methods to simultaneously capture spatial and temporal dependencies often leads to suboptimal prediction outcomes. Fortunately, advancements in the transportation industry have led to the widespread installation of cameras, sensors, and other data collection devices on highways. In terms of temporal correlation, traffic conditions fluctuate across different time periods. For instance, traffic flow tends to be higher during morning, noon, and evening periods compared to other times, while holidays exhibit markedly different traffic patterns compared to regular workdays.Furthermore, a traffic mishap might happen within a specific timeframe, leading to traffic jams. Unavoidably, the traffic volume of the impacted segment in the following time will be affected. Concerning spatial interdependence, when various roads are linked, the traffic condition of one particular road can exert various effects on the adjacent connected roads. To be more specific, upstream roads have the potential to impact downstream ones. Meanwhile, downstream roads can affect upstream roads through feedback.In Figure 1, node A is represented by the purple circle, while its spatially connected nodes B, C, and D are marked by red circles. Here, the blue arrow symbolizes spatial dependence, while the green arrow represents temporal correlation, and yellow arrows indicate spatio-temporal correlations. T and $T+1$ represent two consecutive time steps on the time axis. Specifically, node A influences nodes B, C, and D at time t and also affects itself at time $t+1$. Because of spatiotemporal correlation, node A at time t can even influence on nodes B, C, and D at time $t+1$, as Figure 1.

Classical statistical models gained popularity in their early stages due to their straightforward principles and rigorous mathematical foundations. For example, C.K. Moorthy et al. introduced the ARMA [1] for traffic forecasting. Later, M.Van Der Voort et al. enhanced the ARMA model by incorporating differencing to model time-varying relationships, leading to the creation of the ARIMA [2]. Deep learning has become a popular approach for traffic flow prediction, addressing the shortcomings of traditional statistical methods. SVR models were proposed by M.Lippi and colleagues for traffic forecasting. N. Zarei and team introduced Random Forest methods for traffic prediction, which have shown promising results in handling complex traffic patterns. Additionally, P.Cai and others proposed the KNN model for short-term traffic forecasting, demonstrating its effectiveness in scenarios requiring rapid predictions. Zhuang and Cao[5] proposed a short-term traffic flow prediction method using a convolutional neural network-bidirectional long short-term memory network (CNN-BiLSTM) with multi-component information. Chai et al. [18] developed a multi-feature fusion short-term traffic flow prediction model based on deep learning. Hu et al. [6] introduced the attention mechanism theory and considered the advantages of LSTM and, while RNN models, they used the LSTM-RNN model to predict short-term traffic flow. A hybrid deep learning model for short-term traffic volume prediction was proposed by Zheng et al. [7], integrating a convolutional-long short-term memory network (Conv-LSTM) with an attention mechanism. Similarly, an urban traffic state prediction method based on a self-attention long short-term memory network (SA-LSTM) was introduced by Yu et al. [9]. However, these methods are limited by slower computational speeds. Moreover, when attempting to capture correlations in long-term sequences, they face risks of gradient vanishing or explosion. Additionally, they are limited to capturing temporal correlations and fail to incorporate spatial information from traffic data, resulting in poorer prediction performance. These methods fail to consider the topological structure of the road network, resulting in insufficient exploitation of spatial relationships across regions.

Consequently, spatiotemporal models have gradually emerged, opening up a new chapter in traffic flow prediction. These models are dedicated to deeply integrating the temporal and spatial dimensions, uncovering the complex dependencies hidden within. Dai et al. [3] proposed a method for short-term traffic flow prediction in urban road sections, combining spatiotemporal analysis with GRU. Lv et al. [4] developed a temporal multi-graph convolutional network, which captures spatial, temporal, and semantic relationships among global road network features, providing a broader perspective for traffic flow prediction. Zhao et al. [5] designed a Temporal Graph Convolutional Network (T-GCN), using graph GCN to extract spatial features of traffic flow and GRU to model temporal dependencies. Zhang et al. [26] introduced a traffic volume prediction model using a spatio-temporal convolutional graph attention network, combining graph attention and convolutional neural networks to better capture spatio-temporal traffic flow characteristics. Do et al. [8] utilized spatio-temporal attention to capture spatial features between road sections and temporal dependencies between time steps. The STDSGNN [22] integrates multi-head graph attention and full convolution to model spatial-temporal correlations. Wang et al. [28] proposed the TBC-GNODE model, addressing shallow GNN limitations by using TBC for temporal and GNODE for spatial dependencies, extracting long-range spatio-temporal dependencies through parallel and residual structures. Spatial-temporal correlations are captured by STGODE [21] through the use of tensor-based ordinary differential equations. These advancements in spatio-temporal neural networks, capable of modeling complex dependencies, have become essential for tasks like travel time estimation, traffic demand prediction, and traffic speed and flow prediction [10], [12], [13]. Spatio-temporal models model data using spatial and temporal dimensions. Their excellent performance has received more and more attention. Spatiotemporal graph modeling, owing to its remarkable versatility, has been extensively applied to address various time series prediction challenges, including traffic forecasting. At present, many traffic prediction models have been developed [14], [15], [16].

In the early research on traffic flow forecasting related to spatial dependencies, researchers relied on specific prior knowledge, such as the distances between road segments and the similarity of Points of Interest (POI), to construct graph structures that represent spatial correlations. This approach allowed researchers to quantify the similarity between nodes using Diffusion Convolutional Recurrent Neural Networks (DCRNN) [17] and Spatial-Temporal Graph Convolutional Networks (STGCN) [19], and to construct an adjacency matrix based on this, thereby defining the topological structure of the graph. However, the adjacency matrix generated through this approach lacks a direct link to the forecasting task. Its performance is heavily influenced by the accuracy of prior knowledge and the logical soundness of the construction method, thereby restricting the model's capacity to capture spatial correlations.

To address this limitation, researchers have started exploring adaptive adjacency matrices. Graph WaveNet [20] and the Coupled Layer-wise Convolutional Recurrent Neural Network (CCRNN) leveraged adaptive adjacency matrices to enhance spatial feature extraction, achieving significant success, although they still required predefined graph structures for optimal results. The Adaptive Graph Convolutional Recurrent Network (AGCRN) [10] and the Multi-Task Spatio-Temporal Graph Neural Network (MTGNN) [20] further improved the adaptive adjacency matrix, eliminating the reliance on predefined matrices while delivering comparable performance. However, both predefined and adaptive matrices share a significant limitation: their weights remain static. Traffic conditions are dynamic, with spatial correlations that change over time. The traffic network undergoes complex changes, which predefined graph structures or adaptive matrices are unable to fully represent.

To address these issues, a Flow Attention-based Dynamic Graph Convolutional Recurrent Network (FADGCRN) is proposed. Our approach, which requires no prior knowledge, integrates spatio-temporal embeddings derived from traffic signal time data with dynamic signals extracted from traffic signals. This combination is used to generate a dynamic graph and modify the mapping method to KAN. The principle of flow conservation is incorporated into the attention mechanism, enabling a natural competitive mechanism to be established without the need for specific inductive biases. This distinctive competitive mechanism efficiently resolves the issue of quadratic complexity issue associated with computing attention weights in traditional attention mechanisms. The integration of a time series model and multi-head flow attention enables the model to improve its predictive performance, as it captures a wide range of temporal changes and long-term dependencies. The RNN-based dynamic graph convolutional module (FADGCRU) is employed to extract the spatio-temporal dependencies of traffic signals, utilizing depthwise separable convolution.The key contributions of this research are as follows:

(1) A time-series flow attention mechanism is introduced, replacing the traditional attention weight calculation module with a novel source competition mechanism and sink allocation mechanism, thereby eliminating the influence of specific inductive biases. The use of LSTM networks instead of linear transformations in the attention mechanism enhances the model's ability to capture temporal correlations in long time series.

(2) In the dynamic graph generation process, KAN is used instead of MLP network, and by redefining the role and operation of the activation function, KAN can adaptively adjust the parameters and operational modes of the function based on the data characteristics and learning progress, thereby more effectively capturing complex patterns and relationships within the data.

(3) The integration of combining the flow attention mechanism with RNN can better capture the time-dependent relationship of time series data, adapt to dynamic changes, and generate richer feature representations at the same time, enhancing the model's generalization capability; Additionally, this integration enhances model efficiency and interpretability.

(4) Experiments on four authentic traffic datasets confirm that our model consistently surpasses the benchmark.

The subsequent is the structure of this article:

Part I: We present the research focus of this article, discuss the existing challenges in the field of traffic prediction, and introduce the background and significance of the research.

Part II: We review existing related research work, analyse their strengths and limitations, and point out the advantages of the method in this article in overcoming these disadvantages.

Part III: We introduce in detail the model structure and methodology proposed in this article.

Part IV: We validate the proposed method's effectiveness through extensive experiments and compare it with existing models.

Part V: We summarize the study's findings, discuss its limitations, and suggest future research directions.

II. Related work

A. Traffic flow prediction

Traffic prediction, a thoroughly explored field in intelligent transportation systems, can be regarded as a spatio-temporal forecasting task. Real-time data collected by road sensors is employed to predict traffic conditions. These technologies play a vital role in advancing smart city transportation systems. Initial research in traffic prediction primarily depended on conventional statistical approaches, including HA, ARIMA, and VAR. Nevertheless, these methods require extensive feature engineering and struggle to handle highly nonlinear traffic data, leading to limited predictive accuracy. However, these methods have specific assumptions for data. But in reality, complex traffic data often can not meet these assumptions, resulting in unsatisfactory performance in practical applications. Subsequently, the vector autoregression model in machine learning was introduced into the field of traffic prediction. It can model more complex data. However this method requires elaborate feature engineering and can not handle traffic data with highly nonlinear relationships. At the same time, it can only capture temporal information and ignore the complex correlations between time series. With the breakthroughs of deep learning in other fields, it has been applied to spatio-temporal data prediction, especially in traffic flow prediction. For instance, Zhang et al. [23] constructed the ST-ResNet model based on residual convolutional units, focusing on the spatio-temporal prediction of crowd flow. Yao et al. proposed a model that combines a CNN and LSTM Network, aiming to jointly model the spatio-temporal dependencies of traffic data. Yao et al. [24] also further proposed a spatio-temporal dynamic network that can dynamically learn the similarities between different locations for taxi demand prediction. However, the input data of these models must be standard two-dimensional or three-dimensional grid data, which to some extent limits their application scenarios. The GCGRU model by Xu et al. [25] uses Graph Neural Networks and Recurrent Neural Networks to capture spatial dependencies and temporal correlations in traffic data, enhancing prediction accuracy.

B. Graph convolutional network

In traffic flow prediction, traditional CNNs struggle with non-Euclidean graph-structured data, while GCNs excel. Niepert et al. [27] proposed a heuristic linear method for selecting node neighborhoods in GCNs, achieving effective predictions. Li et al. [13] developed DCRNN, modeling traffic flow as a diffusion process to capture spatial dependencies in directed graphs. Yu et al. [16] introduced STGCN, using graph convolution for spatial dependencies and 1D convolution for temporal correlations, ensuring fewer parameters and faster training. Wu et al. [29] created Graph WaveNet, leveraging an adaptive dependency matrix and stacked 1D-CNNs to enhance spatiotemporal modeling. Bai et al. [10] proposed AGCRN, featuring adaptive modules for node-specific patterns and dynamic spatial dependencies. Song et al. [19] introduced STSGCN, focusing on local spatio-temporal correlations but neglecting global relationships. Tan et al. [30] developed STGPCN, generating cross-spatiotemporal graphs to capture complex node interactions.

C. Attention mechanism

In the field of transportation, attention mechanisms are extensively utilized. These mechanisms filter critical information by assigning varying weights to input data, enhancing computational efficiency, handling variable-length sequences, and enabling the capture of long-term dependencies. They play a crucial role in applications such as speech recognition and traffic flow prediction. Xu et al. [31] developed an

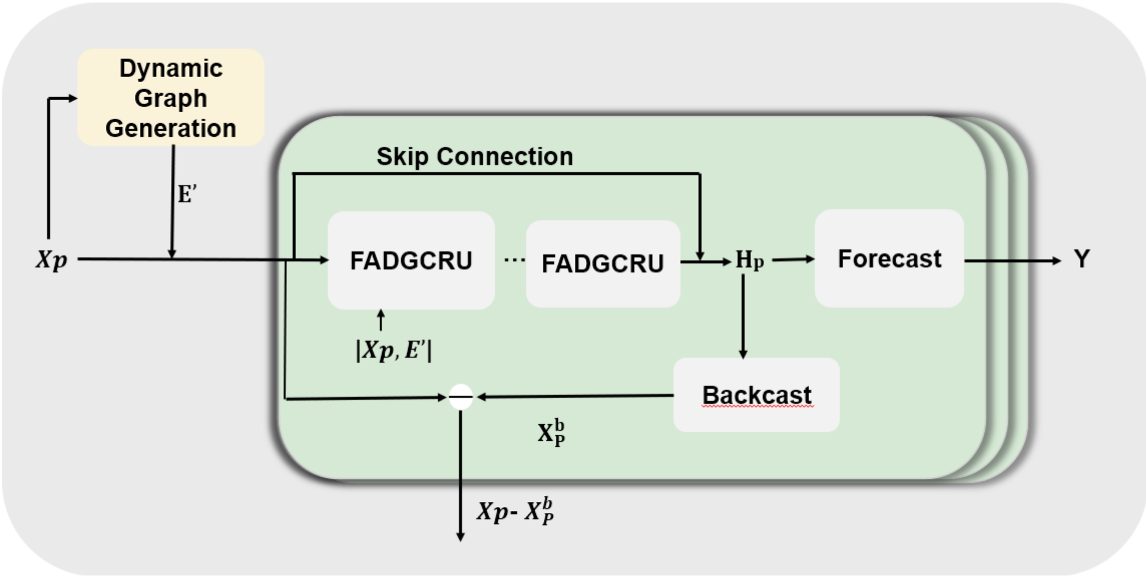


Fig. 2. The architecture of FADGCRN

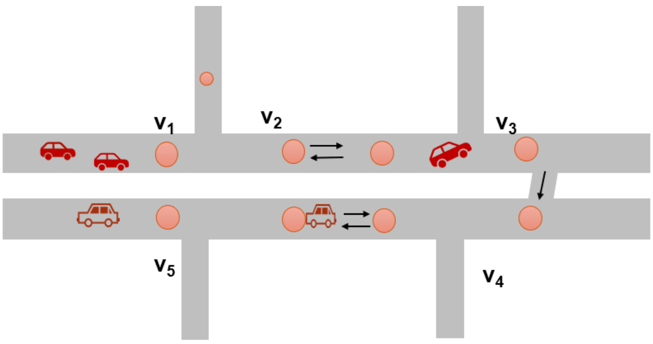


Fig. 3. Road network structure

image caption generator employing both "soft" deterministic and "hard" stochastic attention mechanisms, validating its efficacy through visual analysis. Velickovic et al. [32] merged graph convolution with self-attention to create the Graph Attention Network, achieving notable success in processing graph-structured data. Guo et al. [19] introduced the ASTGCN model, which incorporates a novel spatiotemporal attention mechanism to enhance traffic flow prediction accuracy. Li et al. (as cited in [33]) put forward Detector-Net, which incorporated a multi-view time attention module along with a dynamic attention module. These modules were designed to model both the long-term and short-term temporal dependencies. Zhang et al. [34] designed a spatial self-attention mechanism, integrating it with the Chebyshev network and LSTM to form the SACRN model, aimed at capturing intricate spatiotemporal relationships. Wu et al. [35] integrated the concept of flow conservation into the attention mechanism, introducing the flow attention mechanism, which enhances model universality and reduces computational overhead. Despite these advancements, traditional attention mechanisms and their variants still face challenges in capturing complex spatiotemporal relationships. When applied to large-scale traffic data, there is a need to improve the model's generalization capability and computational efficiency. Further studies may investigate incorporating the flow attention mechanism with time models and graph convolutional networks to better capture comprehensive spatio-temporal correlations. By refining and retaining the multi-head attention approach, the model's expressive power can be further enhanced, leading to superior performance in complex traffic flow prediction tasks.

III. Preliminaries

In this part, the foundational concepts of transportation networks and traffic signals are presented, together with the prediction challenges that remain to be resolved.

Definition 1.**Traffic Sensor**. A traffic detector is a gadget installed within a traffic infrastructure, such as a roadway network. It registers data associated with traffic, including the throughput of passing motor vehicles and their speeds.

Definition 2.**Traffic Network**. The traffic network can be depicted as a graph $G = (V, E, A)$. Here, V is a set encompassing N nodes, each denoting a sensor situated at specific spots in the road network. These sensors are assigned the task of gathering traffic - relevant data at their respective positions. E represents the edge set, and A is a graph formulated based on elements like the distances between pairs of nodes in the network. Our framework makes use of a dynamic adjacency matrix.

Definition 3.**Traffic Forecasting**. Given the historical traffic readings $X_P = [x_{t-P+1} \cdots x_{t-1}, x_t]$ belonging to the space $R^{P\times N\times C}$, the aim of traffic prediction is to forecast the future traffic readings $Y_Q = y_t, y_{t+1} \cdots y_{t+Q}$ which fall within the space $R^{Q\times N\times C}$.

IV. Model framework

This section provides the particulars of FADGCRN. As illustrated in Figure 2, the FADGCRN framework comprises a Dynamic Graph Generation module designed to automatically capture dynamic spatial dependencies. Along with the input data, it is fed into the FADGCRU module, which incorporates skip connections and reverse prediction mechanisms, enhancing the flow of information and allowing the model to better integrate features from different levels, thereby improving overall performance.

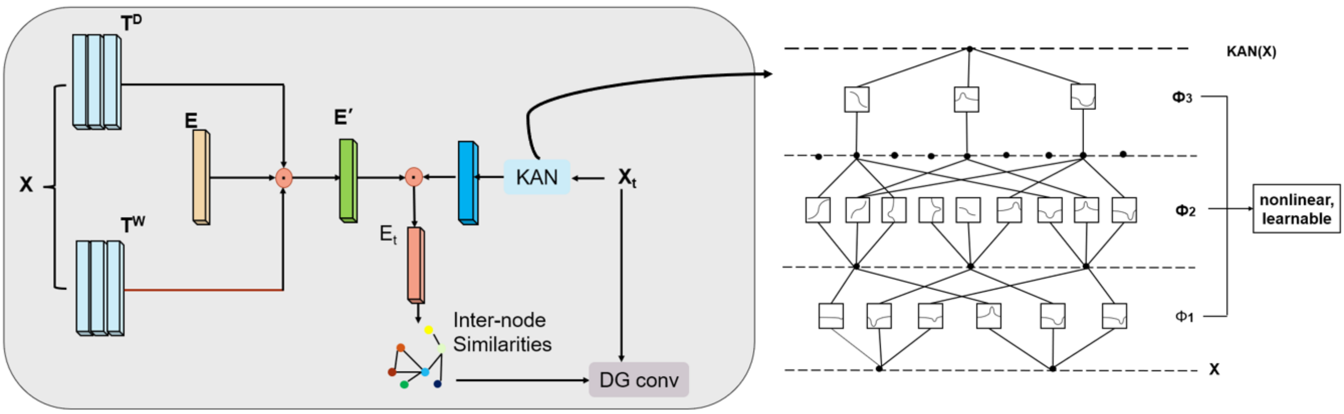


Fig. 4. Dynamic graph generation based on KAN

A. Dynamic graph generation based on KAN

The FADGCRN framework incorporates a dynamic graph generation method, as depicted in Figure 4, which effectively captures dynamic spatial dependencies in sensor data. Consequently, this approach does not depend on a predefined adjacency matrix. In the construction of adjacency matrices, most existing studies rely on the geographical distance between traffic nodes or dynamical model similarity functions. However, the predefined adjacency matrices based on geographical distance do not always accurately reflect the true associations between nodes. As shown in Figure 3 , which shows part of the road network structure, with orange dots representing different traffic nodes. Although Figure 3 shows that node v1 is geographically close to node v5, in reality, the association between v1 and v2 is stronger, and even the association between v1 and v4 (which are geographically far apart) exceeds that between v1 and v5. Therefore the predefined adjacency matrices obtained solely from geographical distance calculations are inaccurate. Manually designing predefined adjacency matrices can improve accuracy but it requires customization for each specific area and relies on the knowledge of domain experts, limiting its generalization capability. Additionally, constructing adjacency matrices based solely on sequence similarity often fails to fully capture spatial correlations, potentially introducing significant biases.

To comprehensively capture the dynamic spatial dependence of the road network at different time points, we design a Spatiotemporal Embedding Generator (STE Generator). The crucial process of the generator is to find out the daily embedding $T^D$and weekly embedding T^W corresponding to the time of the current traffic signal X_P. Subsequently, element-wise product operations are carried out between these embeddings and the spatial embedding E to produce the new spatiotemporal embedding $E^{'}$, as defined in Equation (1).

$$E' = E \times T^D \times T^W \tag{1}$$

Here, T^D and T^W are the daily and weekly embeddings respectively derived from the time step $P = [t-P+1, ..., t]$. And the symbol represents the element-wise product operation.

At each time step t, the input for that step is processed through a KAN layer to extract dynamic signals, as formulated in Equation (2):

$$F_t = KAN(x_t) \tag{2}$$

The KAN layer is capable of learning complex nonlinear relationships within the data, thereby capturing patterns that change over time. The dynamic graph is then generated as described in Equation (3):

$$E_t = \tanh(F_t \times E^{'}) \tag{3}$$

The relationships between nodes are adjusted by the model based on current dynamic signals, allowing for adaptive changes, and the activation function can enhance sensitivity to temporal changes.

The principle of KAN in the aforementioned formula is as follows: The KAN (Knowledge-Aware Network), and the architecture can be seen in Figure 4. I adopted makes this method more advantageous and is especially suitable for any scenario lacking prior knowledge.

The Kolmogorov-Arnold network (KAN)is an innovative neural network,as in Equation (4).

$$KAN(x) = (\Phi_3 \times \Phi_2 \times \Phi_1)(x) \tag{4}$$

It focuses on the Kolmogorov-Arnold (KAR) representation theorem rather than the typical universal approximation theorem in traditional neural networks. In a nutshell, KAN fundamentally transforms the paradigm of the MLP by redefining the role and operation of activation functions. Unlike the static and unlearnable activation functions in MLP, KAN incorporates univariate functions that serve both as weights and activation functions and are adjusted as part of the learning process, as Figure 4. Its main functions include:

Adaptive Grid: Utilizing B-spline basis functions, the grid is dynamically modified based on the input data distribution to maximize the model's representational capacity.

Spline Weights: Spline weights are incorporated to boost the model's nonlinear capabilities. These weights are determined via curve fitting techniques.

Multi-layer Structure: KAN modules can be stacked to form deep networks by combining multiple KANLinear layers.

Activation Functions: KAN supports various activation functions, with the Sigmoid Linear Unit (SiLU) as the default choice.

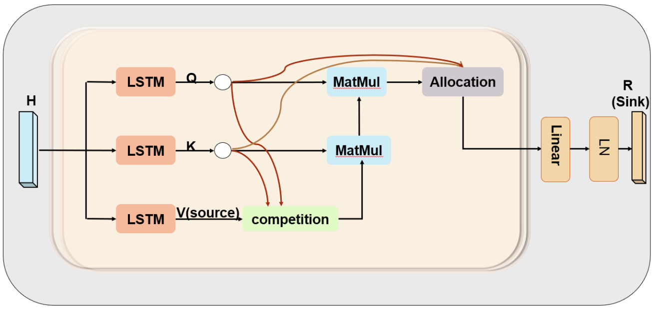


Fig. 5. Temporal Sequence Flow Attention

Regularization: The model incorporates L1 regularization and entropy calculations to control complexity and improve generalization.

In order to fulfill the criteria for Chebyshev polynomials, the dynamic matrix is normalized according to Equation (5):

$$D_t^{-\frac{1}{2}} A_t^d D_t^{-\frac{1}{2}} = D_t^{-\frac{1}{2}} \left(Relu(E_t \times E_t^T) D_t^{-\frac{1}{2}} \right) \tag{5}$$

In Equation (5),the dynamic graph embedding E_t is multiplied by its transpose E_t^T, the interactions and associations between nodes are captured, which allows us to infer the spatial dependencies between nodes.

The dynamic convolution formula is expressed as Equation (6) :

$$\begin{aligned} Z_t = (I_N + D_t^{-\frac{1}{2}} A_t^d D_t^{-\frac{1}{2}}) X \Phi + b = \\ (I_N + D_t^{-\frac{1}{2}} (\text{Relu}(E_t \times E_t^T)) D_t^{-\frac{1}{2}}) X \Phi + b \end{aligned} \tag{6}$$

Subsequently, the GCN is enhanced using the node adaptive parameter learning module. This component allows the model to capture distinct traffic patterns for individual nodes through matrix decomposition. Φ belongs to $R^{N \times C \times F}$. Then, in order to optimize the GCN and prevent overfitting at the same time, the weight matrix is decomposed into a node parameter matrix E_g belongs to $R^{N \times d}$ and two weight matrices W_g $R^{d \times C \times F}$ and bg $R^{d \times F}$, where d ¡= N and $\Phi = E_g W_g$, $b = E_g b_g$, The optimized GCN can be expressed as follows:

$$\begin{aligned} Z_t = (I_N + D_t^{-\frac{1}{2}} A_t^d D_t^{-\frac{1}{2}}) X E_g W_g + E_g b_g = \\ (I_N + D_t^{-\frac{1}{2}} (\text{ReLU}(E_t \times E_t^T)) D_t^{-\frac{1}{2}}) X E_g W_g + E_g b_g \end{aligned} \tag{7}$$

B. Dynamic graph convolution recurrent module based on flow attention

A temporal sequence flow attention mechanism [35] is proposed, which utilizes LSTM to map input data, as shown in Figure 5. Algorithm 7 outlines the training process of the model. Competition among information flows is promoted by the flow attention mechanism through the enforcement of conservation principles at both the source (where data is retrieved from the previous layer) and the sink (where data is forwarded to the next layer). The incoming flow of each sink is normalized to 1, compelling the outgoing flows from the source to compete based on their unique spatial arrangements. This method guarantees efficient information transfer and strengthens the model's ability to model complex dependencies. Similarly, the source's outgoing flow is fixed to 1, forcing the sinks to compete for the single available flow. Once Q, K, and V are obtained via LSTM mapping, conservation is maintained at both the source and sink through the normalization of Q and K, respectively, as illustrated in Figure 5. For a system with n sinks and m sources, the normalization processes are defined in Equations (8) and (9).

$$\frac{\varphi(Q)}{I} \tag{8}$$

$$\frac{\varphi(K)}{O} \tag{9}$$

Here, φ is defined as a non-negative nonlinear function, with I and O representing the inflow of sinks and the outflow of sources, respectively. Equation (8) indicates sink conservation, and Equation (9) indicates source conservation. Normalization ensures that the outflow from the source and the inflow to the sink are balanced, maintaining the stability of the information flow. The detailed steps for this process are outlined in Equation (10) and Equation (11), which provide a clear framework for achieving conservation in the system.

$$I' = \varphi(Q) \frac{\sum_{j=1}^{m} \varphi(K_j)^T}{O_J} \tag{10}$$

$$O' = \varphi(K) \frac{\sum_{i=1}^{n} \varphi(O_i)^T}{I_i} \tag{11}$$

The flow-attention is composed of three key components: competition V', aggregation A, and allocation R, as described by the following equations:

$$V' = softmax(O') \times V \tag{12}$$

$$A = \frac{\varphi(Q)}{I} (\varphi(K)^T V') \tag{13}$$

$$R = LayerNorm(sigmoid(I') \times A + H) \tag{14}$$

The module's input is denoted by H. An advanced reweighting process, which follows the conservation of input flow, promotes competition among information flows. Information aggregation is performed according to the associative properties of matrix multiplication. By filtering the incoming flow for each sink, the allocation mechanism generates the module's final output. To address network degradation and minimize the risk of gradient vanishing, we incorporate a residual mechanism combined with layer normalization.

In our experiments, GRUs within RNNs exhibited remarkable ability in capturing both temporal and spatial dependencies. By introducing the DG conv(implementation of the dynamic graph convolutional network from the previous section), graph convolution operations are incorporated, and a flow attention mechanism is introduced, enabling the model to consider the connections and interactions between nodes when updating states, as depicted in Equation (6). The Flow-Attention-DGRU can be expressed as follows.

$$r_t = \sigma(\theta[x_t || H_{t-1}, E_t] E W_r + E b_r) \tag{15}$$

$$u_t = \sigma(\theta[x_t || H_{t-1}, E_t] E W_u + E b_u) \tag{16}$$

$$\hat{h}_t = tanh(\theta[x_t || r_t \times H_{t-1}, E_t] E W_c + E b_c) \times A_t \tag{17}$$

$$H_t = \mu_t \times H_{t-1} + (1 - \mu_t) \times \hat{h}_t \tag{18}$$

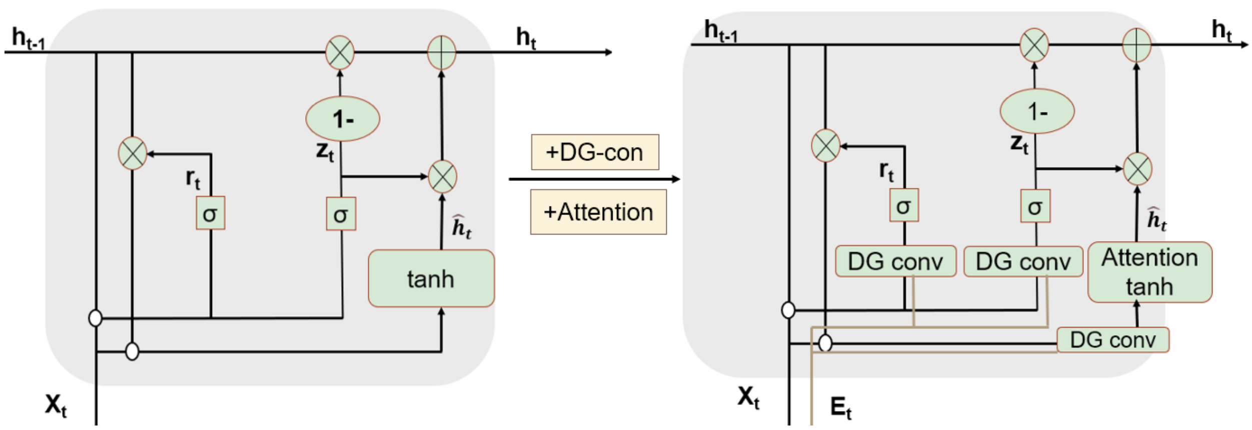


Fig. 6. The model of FADGCRU

Here, x_t and H_t represent the input and output at time step t. The output of the attention mechanism is denoted by A_t. The sigmoid activation function is represented by σ. The dynamic graph generation is indicated by θ. The concatenation operation is symbolized by $||$. The learnable parameters include W_r, W_u, W_c, b_r, b_u, and b_c.

Backcasting, as illustrated in Figure 2, aids in analyzing historical data and patterns. It offers insights into the system's past evolution and can be utilized to inform predictions. Following FADGCRU, there is an output sub-layer composed of linear layers. This sub-layer produces two outputs: $\hat{y} = \text{Linear}(H_p)$, $x^b = \text{Linear}(H_p)$,$x_p - x_p^b$ indicates that the learned information has been removed. Each prediction step is adjusted based on the outcomes of the preceding step, enhancing the accuracy and stability of multi-step forecasting. Finally, the multi-step prediction results are added.

Algorithm 1 Training the FADGCRN model

1: **Input:** N, T, T', X, traffic graph features, training epochs;
2: **for** epoch = 0 to epochs - 1 **do**
3: Randomly initialize learnable parameters of FADGCRN;
4: Generate a dynamic graph embedding: $E_t = \tanh(F_t \times E')$;
5: Generate the dynamic graph convolution: $Z_t = (I_N + D_t^{-\frac{1}{2}} A_t^d D_t^{-\frac{1}{2}}) X\Phi + b = (I_N + D_t^{-\frac{1}{2}} (\text{ReLU}(E_t \times E_t^T)) D_t^{-\frac{1}{2}}) X\Phi + b$
6: Combine the dynamic graph convolution method with the RNN module and replace the matrix product in GRU with the normalized recurrent unit of the previous step's dynamic graph convolution;
7: Use the flow attention mechanism to adjust weights output H_t;
8: Skip connection and multi-step prediction $\hat{y} = \text{Linear}(H_p)$;$x^b = \text{Linear}(H_p)$
9: Compute loss $\mathcal{L} = \text{loss}(\hat{Y}, Y_{\text{true}})$;
10: Optimize trainable parameters using gradient descent;
11: **end for**
12: **Output:** the learned FADGCRN model;

V. Experimental evaluation

A. Datasets

We evaluated our model's effectiveness by conducting comparative experiments on four real-world highway traffic datasets, as detailed in TABLE I. We split the datasets into training, validation, and test sets in a 6:2:2 ratio.

1) Data: The PEMSD3 dataset, gathered by the Caltrans Performance Measurement System (PeMS), comprises data from 358 sensors spanning September to November 2018. Data is captured every 5 minutes, yielding 26,208 time slices.

The PEMSD4 dataset, sourced from CalTrans PeMS, encompasses data from January 1, 2018, to February 28, 2018, including 307 sensors. The data is recorded every 5 minutes, producing 16,992 time slices.

The PEMSD7 dataset, sourced from PeMS, encompasses data from May to August 2017, including 883 sensors. The data is recorded every 5 minutes, producing 28,224 time slices.

The PEMSD8 dataset, sourced from CalTrans PeMS, encompasses data from July 1, 2018, to August 31, 2018, including 170 sensors. The data is recorded every 5 minutes, producing 17,833 time slices.

2) data normalization method: Standard normalization usually normalizes data to an interval with a mean value of 0 and a standard deviation of 1. The formula is given as follows:

$$x_{normalized} = \frac{x - \mu}{\sigma}$$

In this context, x is used to denote the original data value, μ is defined as the mean of the data, and σ is identified as the standard deviation of the data. Through standard normalization, data of different features can have a similar scale, which helps improve the convergence speed and prediction ability of the model. Because when the scale differences of data are large, the model may be more inclined to learn features with larger numerical values and ignore features with smaller numerical values. Normalization ensures a more balanced importance of each feature in model learning, enhancing the model's performance. In traffic flow prediction, the normalized traffic flow value obtained through model prediction needs to be inversely normalized to the actual traffic flow value before actual analysis and decision-making can be carried out. The formula is as follows:

$$x_{original} = x_{normalized} \times \sigma + \mu$$

TABLE I
DATASETS DETAILS

Datasets	Nodes	Rate	Time steps	Time Range
PEMSD3	358	26208	5min	2018.09 - 2018.11
PEMSD4	307	16992	5min	2018.01 - 2018.02
PEMSD7	883	28224	5min	2017.05 - 2017.08
PEMSD8	170	17856	5min	2016.07 - 2016.08

where $x_{original}$ is the original data value after inverse normalization,$x_{normalized}$ is the data value after standard normalization, and is the mean of the original data.

B. Experimental settings

The model is implemented in PyTorch 1.9.1, with computations accelerated on an NVIDIA 3090 GPU. For the PEMS04 dataset, AdamW is chosen as the optimizer, with the learning rate set to 0.007 and a Multi-Step Learning Rate Scheduler used, along with a batch size of 32. For PEMS08, AdamW is similarly selected, with a learning rate of 0.008 and a cosine decay schedule applied, and a batch size of 32. In the case of PEMS03, AdamW is utilized, with the learning rate fixed at 0.009 and a Multi-Step Learning Rate Scheduler employed, with a batch size of 64. For PEMS07, AdamW is applied, with a learning rate of 0.001 and a cosine decay schedule implemented, and a batch size of 16. Overfitting is prevented through the use of early stopping and regularization techniques. All experimental results are subjected to significance tests (t-tests with $p<0.05$).

All baseline models are assessed using three widely recognized traffic prediction metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics enable a thorough evaluation of the models' accuracy and robustness. The model's performance is measured using three key indicators:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i| \tag{19}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{20}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \hat{Y}_i|}{|Y_i|} \times 100 \tag{21}$$

C. Baselines

We have chosen a wide range of baseline models with publicly available official codes, encompassing traditional approaches, typical deep learning techniques, and the latest state-of-the-art methods.

ARIMA [36]: This model integrates autoregressive and moving average components to forecast future values by fitting time series data.

HA [37]: The model treats traffic flow as a cyclical process and forecasts future time intervals future periods using the weighted average of past periods.

VAR [37]: Vector autoregression assumes stationarity in the time series and estimates relationships between the series and its lagged values.

FC-LSTM [38]: The fully connected LSTM network is a well-known architecture, highly effective at capturing sequential dependencies.

DCRNN [17]: The fully connected layer in GRU can be replaced by a diffusion convolutional layer. As a result, a new type of diffusion convolutional gated recurrent unit is formed.

AGCRN [10]: AGCRN employs an adaptive graph and a GRU to capture spatio-temporal features.

Graph WaveNet [29]: This model layers gated TCN and GCN to simultaneously capture both spatial and temporal dependencies.

ASTGCN [19]: This approach integrates spatiotemporal attention mechanisms to effectively model the dynamic spatiotemporal characteristics of traffic data.

STSGCN [19]: Designed to efficiently model local spatiotemporal relationships, STSGCN also addresses the heterogeneity of spatiotemporal data.

STFGNN [39]: The spatiotemporal fusion graph neural network integrates multiple spatiotemporal graphs to effectively capture complex spatiotemporal correlations, enhancing the model's ability to handle dynamic traffic patterns.

STSGRU [40]: This model introduces a spatial–temporal shared GRU, designed to capture long-term temporal characteristics by examining weekly traffic trends. It employs a shared weight mechanism to support predictions across multiple scenarios.

LEISN-ED [41]: This network features a module dedicated to long-term dependencies, enhancing the transfer of extended temporal features. It also integrates dual spatial feature extraction pathways to independently gather both explicit and implicit spatial attributes.

DSTAGNN [42]: The dynamic spatial-temporal aware graph neural network makes use of a sophisticated multi-head attention approach. It is this mechanism that allows the network to model dynamic spatial interactions. Meanwhile, it captures temporal dependencies by carrying out multi-scale gated convolutions on features with multiple receptive fields.

VI. EXPERIMENT RESULTS AND ANALYSIS

A. Performance on PeMS Datasets

TABLES II and III compare the performance of various models across four datasets. Our model consistently outperforms others, achieving the highest accuracy within 12 time steps. Traditional approaches and machine learning techniques, which focus exclusively on temporal relationships and demand data stationarity, are not well-suited for modeling the spatiotemporal features of traffic data. As shown in the tables, these methods perform poorly. The FC-LSTM model leverages deep learning techniques to effectively capture temporal patterns in traffic data, yielding promising results. However, its inability to incorporate spatial relationships limits its overall performance, making graph-based models more accurate. These findings highlight the critical importance of integrating spatial dependencies in traffic data modeling.

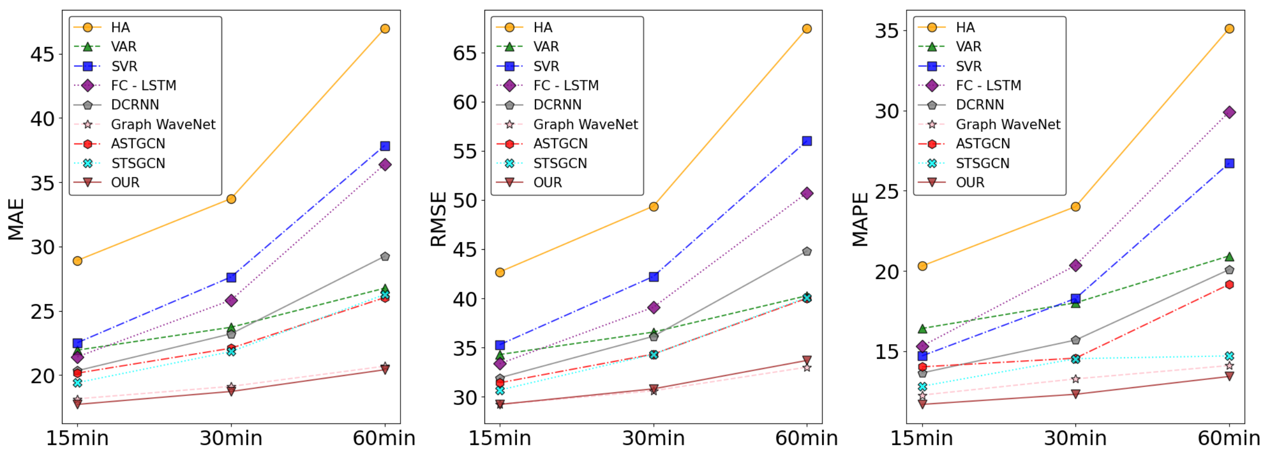
Among graph-based models, Graph WaveNet effectively captures the combined impact of temporal and spatial factors on traffic flow changes, and ASTGCN's attention mechanism

TABLE II
PERFORMANCE COMPARISON OF VARIOUS MODELS ON PEMSD4 AND PEMSD8 DATASETS

Variants	**PEMSD4**			**PEMSD8**		
	MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
HA	38.03	59.24	27.88%	34.86	59.24	27.88%
ARIMA	33.73	48.80	24.18%	31.09	44.32	22.73%
VAR	24.51	38.61	17.24%	19.19	29.81	13.10%
FC-LSTM	26.77	40.65	18.23%	19.19	29.81	13.10%
DCRNN	21.22	33.44	14.17%	17.86	27.83	11.45%
GraphWaveNet	24.89	39.66	17.29%	18.28	30.05	12.15%
ASTGCN(r)	22.92	35.22	16.56%	18.25	28.06	11.64%
MSTGCN	23.96	37.21	14.33%	19.00	29.15	12.38%
STSGCN	21.19	33.65	13.90%	17.13	26.80	10.96%
STFGNN	20.48	32.51	16.77%	16.94	26.25	10.60%
STSGRU	20.11	31.80	13.86%	15.68	25.12	10.67%
LEISN-ED	20.84	32.82	13.77%	16.81	25.97	10.62%
AGCRN	19.83	32.26	12.97%	15.95	25.22	10.09%
DSTAGNN	19.30*	31.46*	12.70*%	15.67*	24.77*	9.94*%
Our model	**18.75**	**30.85**	**12.29%**	**14.61**	**23.71**	**9.51%**

TABLE III
PERFORMANCE COMPARISON OF VARIOUS MODELS ON PEMSD3 AND PEMSD7 DATASETS

Variants	**PEMSD3**			**PEMSD7**		
	MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
HA	31.58	52.39	33.78%	45.12	65.64	24.51%
ARIMA	35.41	47.59	33.78%	38.17	59.27	19.46%
VAR	23.65	38.26	24.51%	50.22	75.63	32.22%
FC-LSTM	21.33	35.11	23.33%	29.98	45.94	13.20%
DCRNN	17.99	30.31	18.34%	25.22	38.61	11.82%
GraphWaveNet	19.12	32.77	18.89%	26.39	41.50	11.97%
ASTGCN(r)	17.34	29.56	17.21%	24.01	37.87	10.73%
MSTGCN	19.54	31.93	23.86%	29.00	43.73	14.30%
STSGCN	17.48	29.21	16.78%	24.26	39.03	10.21%
STFGNN	16.77	28.34	16.30%	23.46	36.60	9.21%
STSGRU	15.45	**24.13**	15.85%	21.50	34.40*	9.08%
AGCRN	15.98	28.25	15.23%	22.37	36.55	9.12%
DSTAGNN	15.57*	27.21	**14.68%**	21.42*	34.51	9.01*%
Our model	**15.13**	26.21*	14.93*%	**20.76**	**34.20**	**8.92%**



(a) MAE on Different Models (b) RMSE on Different Models (c) MAPE on Different Models

Fig. 7. Forecasting Metrics on Different Models on PEMSD4

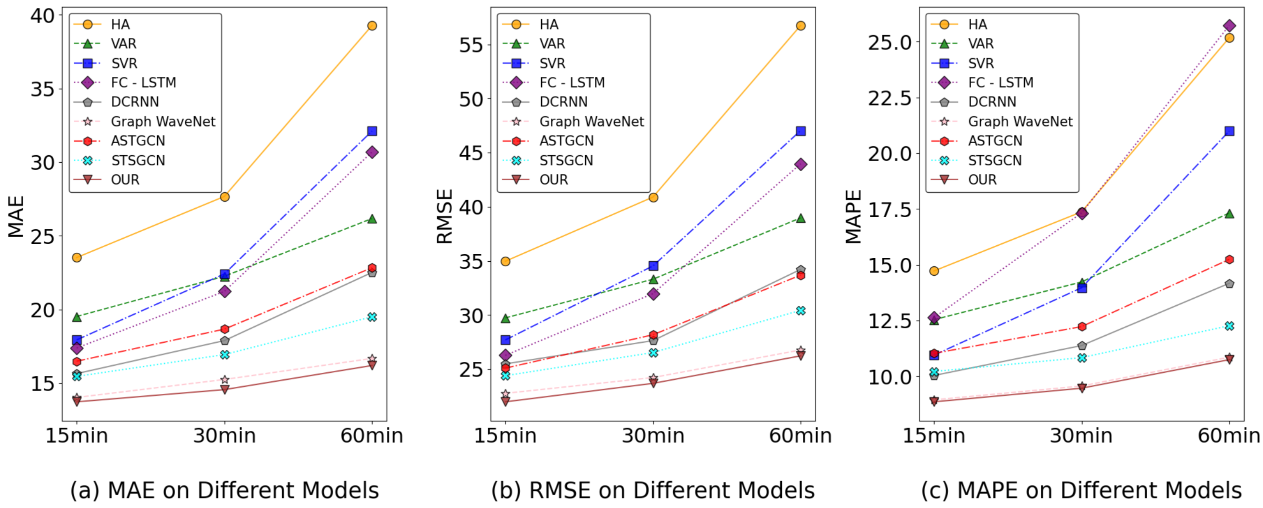


Fig. 8. Forecasting Metrics on Different Models on PEMSD8

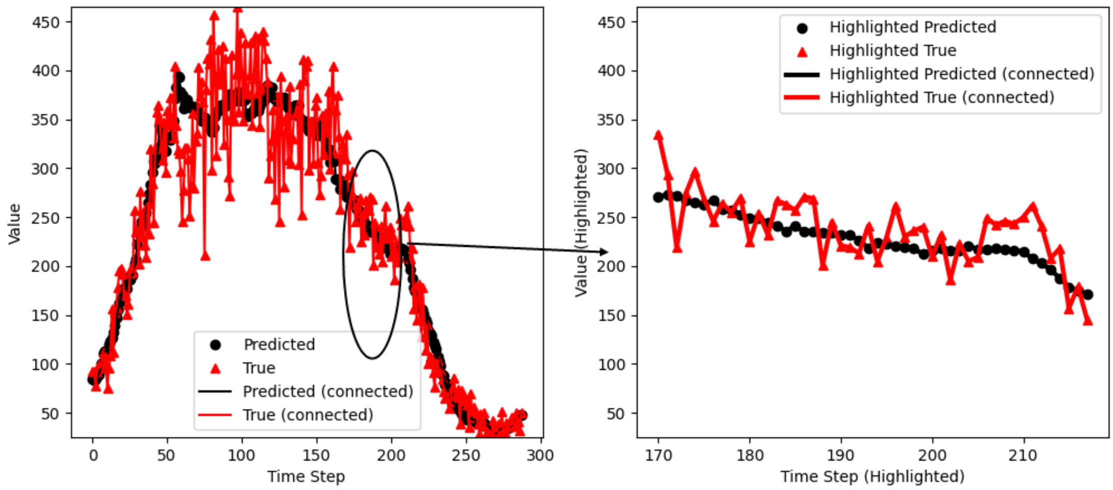


Fig. 9. Prediction Curves in PEMSD4 - Node 10

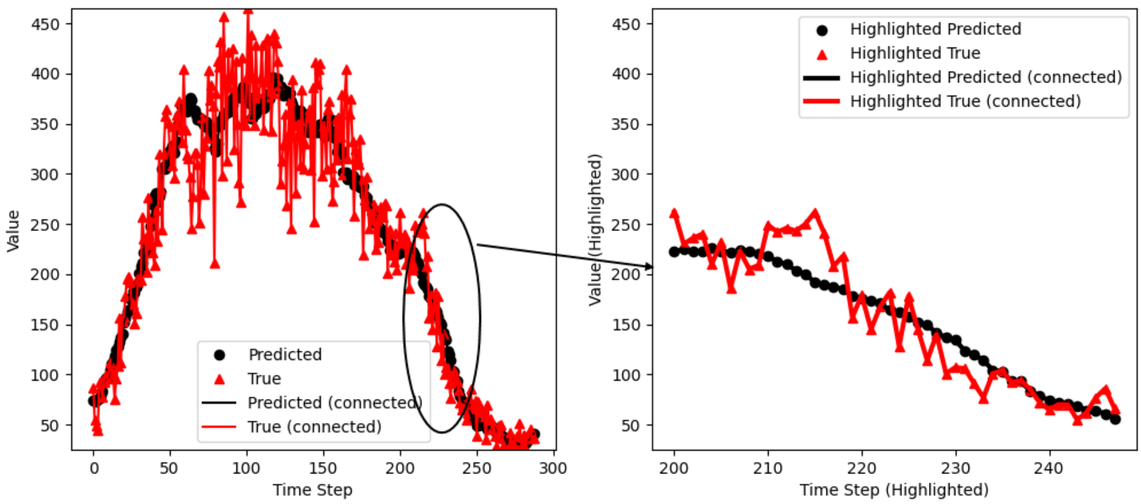


Fig. 10. Prediction Curves in PEMSD4 - Node 166

has strong capabilities in capturing long-term dependencies. Both models demonstrate strong predictive performance. However, the temporal and spatial modeling components in these models are relatively basic. STSGCN overcomes this limitation by employing multiple local spatiotemporal subgraphs to capture the heterogeneity of spatiotemporal data, facilitating a more detailed analysis of interactions within the data. DCRNN, on the other hand, captures spatial correlations through predefined graph structures. Overall, these models achieve competitive predictive results.

TABLE IV
THE ABLATION EXPERIMENTS OF TRAFFIC FORECASTING ON THE PEMSD8 DATASETS.

Variants	Horizon 3			Horizon 6			Horizon 12		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
adaptive graph	14.9895	23.4018	9.5200%	15.9889	25.1729	10.1600%	17.9389	27.9300	11.4700%
normal_attention	13.7118	**21.9705**	8.94%	14.5282	23.6381	9.4900%	16.0739	26.2743	10.9400%
mlp	13.7909	22.0924	8.8800%	14.5840	23.8067	**9.38%**	16.2165	26.3147	**10.6300%**
normal_convolution	13.9500	22.4365	9.03%	15.0233	24.3405	9.6000%	16.7474	27.0659	11.0500%
gru	13.7752	22.1133	8.9800%	14.7108	23.9632	9.6200%	16.3046	26.4615	10.7300%
liner	14.0756	22.6121	9.0300%	14.9848	24.2826	9.5500%	16.5409	26.8044	10.8200%
residual mechanism	13.7459	22.6523	8.7800%	14.6219	23.8923	9.5300%	16.3029	26.4587	10.8200%
final	**13.7022**	21.9770	**8.86%**	**14.5009**	**23.6130**	9.4200%	**16.0395**	**26.2240**	10.7500%

TABLE V
THE ABLATION EXPERIMENTS OF TRAFFIC FORECASTING ON THE PEMSD4 DATASETS.

Variants	Horizon 3			Horizon 6			Horizon 12		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
adaptive graph	18.6468	29.8989	12.5300%	19.6763	31.5302	13.2700%	21.4254	34.1258	14.6300%
normal_attention	17.8720	**29.2138**	11.9000%	18.7983	30.7966	12.44 %	20.2011	33.6561	13.3800%
mlp	18.1188	29.4890	12.2600%	18.9949	30.9338	12.7800%	20.8061	33.6175	14.1100%
normal_convolution	17.8555	29.2578	11.8900%	18.7579	30.7989	**12.4000%**	20.4620	**33.5414**	13.3800%
gru	17.9586	29.3307	12.3100%	18.7498	30.7693	12.8400%	20.2466	32.9706	13.6200%
liner	18.2883	29.6325	12.3300%	19.2083	31.1387	12.8700%	21.0421	33.7626	14.4700%
residual mechanism	17.9034	29.2307	12.3073%	18.7652	30.7372	12.7405%	20.2565	33.1706	13.5216%
final	**17.8182**	29.3974	**10.9900%**	**18.7334**	**30.7652**	12.4100%	**20.1841**	33.8032	**13.3300%**

However, the effectiveness of predefined graph structures significantly influences the model's final performance. LEISN-ED addresses this limitation by employing two graph convolution branches: one extracts explicit spatial features based on adjacency matrices derived from spatial topology, while the other captures implicit spatial features using adjacency matrices generated from trend similarity. Models such as AGCRN further enhance spatial relationship modeling by producing adaptive matrices, achieving superior results. However, the aforementioned graph models still rely on static graph structures, failing to account for the dynamic nature of the data. DSTAGNN utilizes a data-driven dynamic spatiotemporal aware graph approach to develop a new spatiotemporal attention module, which captures dynamic temporal dependencies between nodes through an enhanced multi-head attention mechanism. Additionally, a spatial attention module is employed to refine graph convolution operations, enabling the capture of dynamic spatial dependencies between nodes. This results in the best predictive performance among the selected baselines.TABLES II and III present the one-hour ahead prediction results on four datasets, where the bold numbers denote the optimal outcomes and the numbers with an asterisk(*) signify the second-best results. Compared to other baseline models, FADGCRN, which constructs dynamic matrices using weekly and daily embeddings, consistently achieves competitive results, demonstrating superior accuracy. The Figure 7 and 8 further presents the performance of different models in multi - step prediction on the PEMSD4 and PEMSD8 datasets, covering nine models such as HA and SVR. Through three metrics, the performance of each model under three time horizons of 15 minutes, 30 minutes, and 60 minutes is compared. In multi-step prediction, the best performance is almost consistently achieved by the model proposed in this study.

In Figure 9 and 10, we intercept one day on the test set of PeMSD4 and randomly select two nodes. Then, we draw the 24-hour prediction curve and a clearer 4-hour prediction curve respectively and compare them with the true values.

VII. ABLATION EXPERIMENTS

A. module ablation experiments

The impact of removing or modifying individual components is evaluated through ablation studies, which analyze the system's overall performance. In the context of traffic flow forecasting, these studies are frequently used to identify the significance and influence of individual components within a comprehensive model. To assess the effectiveness of various modules in FAGCRN, we developed six variants of the FADGCRN model. Each variant was tested under optimal conditions. We compared the prediction outcomes of these six variants with FAGCRN on the PeMSD4 and PeMSD8 datasets.

(1)'adaptive graph': Replacing the dynamic graph generation mechanism with the adaptive graph generation mechanism effectively demonstrates the advantage of dynamic graph generation in capturing spatio-temporal dependencies.

(2)'normal attention': The unique advantages of the flow attention mechanism can be effectively verified when the traditional attention mechanism is utilized to replace it.

(3)'mlp': In the process of dynamic graph generation, replacing the KAN with a MLP highlights the significant

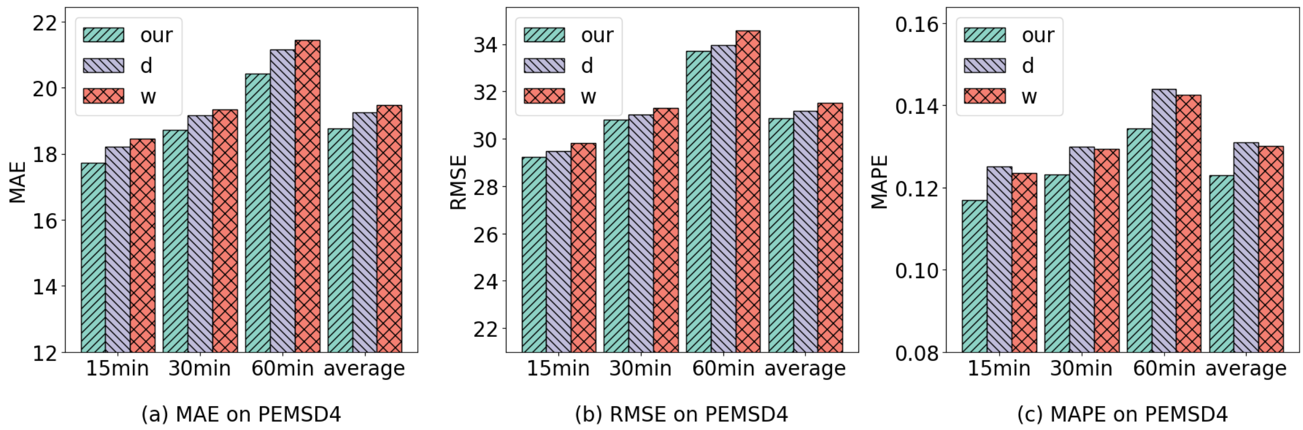


Fig. 11. Ablation experiment of time embedding On PEMSD4

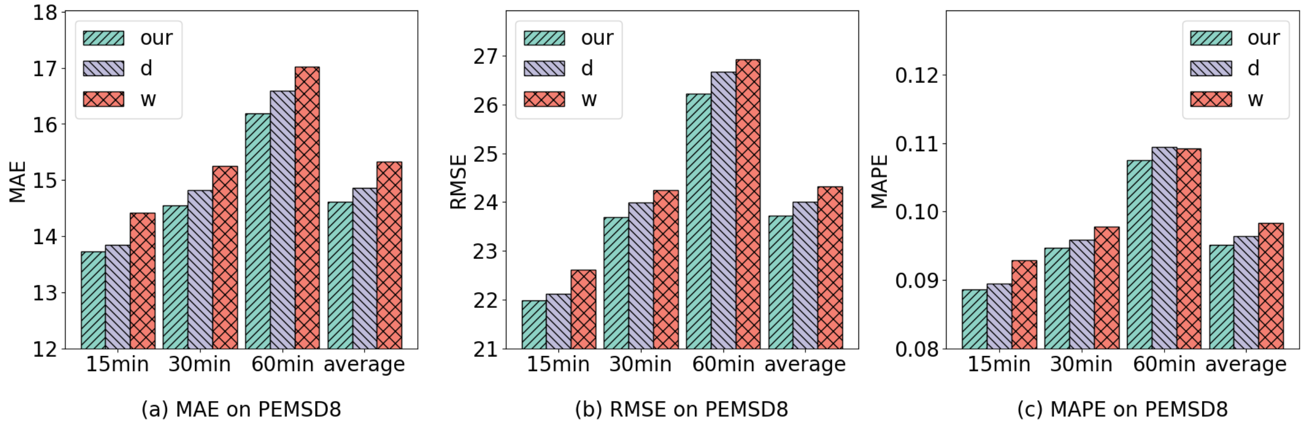


Fig. 12. Ablation experiment of time embedding On PEMSD8

advantages of KAN.

(4)'normal convolution': Replacing the depthwise separable convolution with ordinary convolution successfully verifies that the depthwise separable convolution has stronger data modeling capabilities.

(5)'gru': Substituting LSTM with GRU in the flow attention mechanism demonstrates that LSTM significantly improves the modeling of time series data.

(6)'linear': Using a linear layer instead of LSTM in the flow attention mechanism confirms LSTM's superior ability to model time series data effectively.

(7)'the residual mechanism': Removing the parts related to residual connections in the FADGCRU module and letting the model perform calculations in a conventional feed - forward manner effectively demonstrates the advantages of residual connections.

TABLES IV and V present the results of ablation experiments for six variants of the FADGCRN model on the PEMSD4 and PEMSD8 datasets. Although the evaluation metrics are not the lowest in some ablated modules, the overall experimental results of the model in this study are superior to those of the other variants in the ablation experiments. The precision of this model is significantly enhanced compared to variants using adaptive graphs, linear mappings, conventional convolutions, the residual mechanism or MLPs. This indicates that, after comprehensively considering the advantages of various modules, the model can better predict traffic flow in both long-term and short-term predictions. The "final" version, which employs dynamic graph generation with KAN, LSTM mapping, and depthwise separable convolutions, can more accurately capture the trends in traffic flow changes, providing a more reliable basis for traffic management and planning.

B. Time embedding

To assess the impact of daily and weekly time embeddings, we designed two variants for ablation studies:

Variant D: This variant only uses daily embedding and removes weekly embedding to generate dynamic embeddings.

Variant W: This variant only relies on weekly embedding and removes daily embedding to generate dynamic embeddings.

The experiments utilized the PEMSD4 and PEMSD8 datasets, widely recognized in traffic forecasting research. The average values of all evaluation metrics were calculated and analyzed to assess the performance of each variant. As shown in Figure 11 and 12, the results are presented. In brief, all three types of time embeddings are effective. Meanwhile, using both weekly embedding and daily embedding is the most effective, whether for short-term prediction or long-term prediction.

VIII. Parameter analysis

The impact of two hyperparameters, numbers of heads and embedding dimension, is analysed. In experiments on the

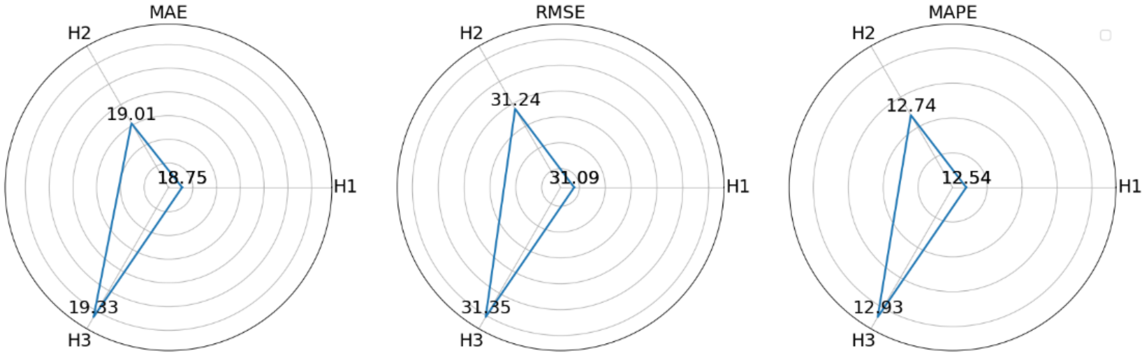


Fig. 13. Sensitivity to numbers of heads on PEMSD4

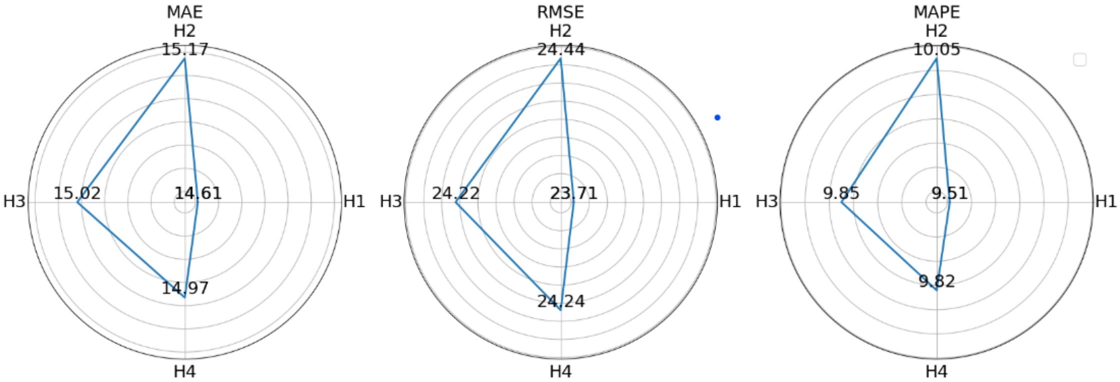


Fig. 14. Sensitivity to numbers of heads on PEMSD8

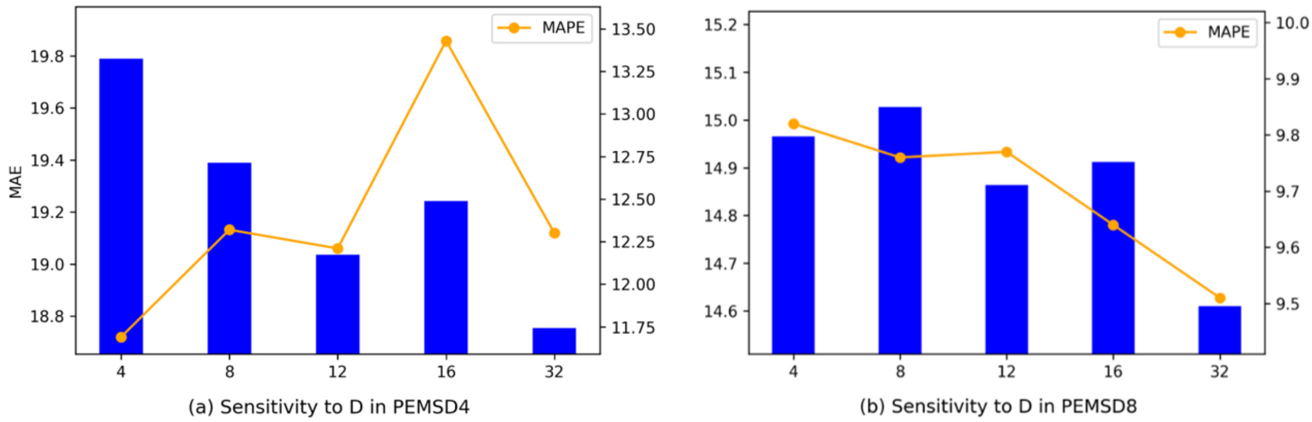


Fig. 15. Sensitivity to D

PEMSD8 and PEMSD4 dataset, we change the value from 1 to 3 on PEMSD4 and 1 to 4 on PEMSD8, and vary the value of among 4, 8, 16, and 32.

A. Sensitivity to numbers of heads

In the multi-head attention mechanism, the article tested 1 head,2 heads, 3 heads, and 4 heads(PEMSD8) respectively. As depicted in Figure 13, 14, among them, H1 has the smallest radius, 1 head has the best effect. First of all, when there is only 1 head, the model can focus more on specific aspects of the input data without being distracted by multiple parallel attention processes, improving the accuracy of encoding. Moreover, a single head is less likely to overfit. When there is a single head, since there are fewer parameters to adjust, the model is easier to generalize to new data.

B. Sensitivity to D

Figure 15 illustrates the impact of varying embedding dimensions on model performance. D represents the embedding dimension of FADGCRN. In this case, the model achieves optimal performance only when D falls within a particular range. This implies that there is an optimal range, often referred to as the "Goldilocks zone," for the embedding dimensionality. If D is too small, the model fails to adequately capture both spatial and temporal information, which results in decreased performance. Conversely, excessively large values of D may lead to overfitting.

IX. Conclusions

This study introduces a dynamic graph convolutional recurrent network augmented with a flow attention mechanism for traffic flow prediction. Comprehensive experiments were carried out on four real-world traffic datasets, and the experimental results are rigorously compared and analysed with the baseline model. The main findings are outlined below:

(1) We introduce a flow-attention mechanism grounded in the "conservation of flow" theory, which adeptly addresses the degradation issues faced by conventional attention mechanisms when processing lengthy sequence data while preserving the model's versatility and expressive power. By incorporating source competition and sink distribution mechanisms, this approach not only mitigates computational complexity but also chieves superior predictive performance. The integration of LSTM's nonlinear feature mapping enhances the mechanism's ability to capture both long-term and short-term dependencies within traffic flow data, with experimental evidence affirming its efficacy in both short-term and long-term forecasting scenarios.

(2) Given the intricate spatio-temporal distribution of traffic flow data, which can exhibit heterogeneity across various regions and time periods, we propose a dynamic graph generation framework that leverages daily and weekly embeddings. This method more effectively captures the global temporal and local spatial relationships between nodes, significantly bolstering predictive accuracy. The adaptive grid of KAN, capable of dynamic adjustment in accordance with the distribution of traffic flow data, better accommodates the data's characteristics, thereby enhancing the network's expressive capabilities. Consequently, the integration of the KAN module with GCN in this paper effectively uncovers the latent spatio-temporal correlations within traffic flow, facilitating efficient feature information propagation and further elevating the model's performance.

(3) Traffic flow data, being a quintessential time series, is adeptly handled by GRU, which excels at capturing sequential dependencies and retaining historical information. The flow attention mechanism further intensifies the focus on critical time points or intervals, enabling the model to more precisely comprehend the evolution of traffic flow and adapt to its dynamic fluctuations.

(4) Ablation experiments conducted in this study validate the critical role each key module plays in the traffic flow prediction task. The experimental outcomes reveal that the model's predictive performance on two datasets significantly surpasses that of the baseline model, thereby fully demonstrating the model's robust generalization and predictive capabilities.

Future research directions:

(1) Model Optimization: Future work could explore advanced structural algorithms, such as fusing the advantages of multiple deep learning models, incorporating reinforcement learning to optimize the parameter decision-making process or exploring more effective attention mechanism models. At the same time, study the online update and adaptive adjustment of the model to adapt to the real-time changes of traffic.

(2) Multi-source Data Fusion: Future research could focus on integrating multiple data sources(like weather, road events, social media data, etc.), integrate and process multi-source heterogeneous data, and mine associated complementary information to improve the prediction ability and adaptability.

REFERENCES

[1] Moorthy, C.K. and Ratcliffe, B.G., "Short Term Traffic Forecasting Using Time Series Methods," Transportation Planning and Technology, vol. 12, pp. 45–56, 1988.

[2] Van Der Voort, M., Dougherty, M. and Watson, S., "Combining Kohonen Maps with ARIMA Time Series Models to Forecast Traffic Flow," Transportation Research Part C: Emerging Technologies, vol. 4, pp. 307–318, 1996.

[3] G. Dai, C. Ma, X. Xu, "Short-Term Traffic Flow Prediction Method for Urban Road Sections Based on Space-Time Analysis and GRU," in *IEEE Access*, vol. 7, pp. 143025–143035, 2019.

[4] M. Lv, Z. Hong, L. Chen, T. Chen, T. Chen, T. Zhu, S. Ji, "Temporal Multi-Graph Convolutional Network for Traffic Flow Prediction," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 3337–3348, 2021.

[5] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, H. Li, "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 3848–3858, 2020.

[6] H. Hu, Z. Lin, Q. Hu, Y. Zhang, "Attention Mechanism With Spatial-Temporal Joint Model for Traffic Flow Speed Prediction," in *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2021.

[7] H. Zheng, F. Lin, X. Feng, Y. Chen, "A Hybrid Deep Learning Model With Attention-Based Conv-LSTM Networks for Short-Term Traffic Flow Prediction," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 6910–6920, 2021.

[8] L.N.N. Do, H.L. Vu, B.Q. Vo, Z. Liu, D. Phung, "An effective spatial-temporal attention based neural network for traffic flow prediction," in *Transportation Research Part C: Emerging Technologies Symposium*, vol. 108, pp. 12–28, 2019.

[9] J. Yu, H. Wei, H. Guo, Y. Cai, "Urban Traffic State Prediction Based on SA-LSTM," in *IOP Conference Series: Earth and Environmental Science*, vol. 783, pp. 012153, 2021.

[10] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 17 804–17 815.

[11] W. Zhuang, Y. Cao, "Short-Term Traffic Flow Prediction Based on CNN-BILSTM with Multicomponent Information," in *Applied Sciences Conference Proceedings*, vol. 12, 2022.

[12] S. Cao, L. Wu, J. Wu, D. Wu, and Q. Li, "A spatio-temporal sequence-to-sequence network for traffic flow prediction," *Information Sciences*, vol. 610, pp. 185–203, 2022.

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

[14] Y. Chen, I. Segovia, and Y. R. Gel, "Z-gcnets: time zigzags at graph convolutional networks for time series forecasting," in *International Conference on Machine Learning*, ser. PMLR, 2021, pp. 1684–1694.

[15] J. Choi, H. Choi, J. Hwang, and N. Park, "Graph neural controlled differential equations for traffic forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 6367–6374.

[16] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

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

[18] C. Chai, C. Ren, C. Yin, H. Xu, Q. Meng, J. Teng, G. Gao, "A Multifeature Fusion Short-Term Traffic Flow Prediction Model Based on Deep Learnings," in *Journal of Advanced Transportation Symposium*, vol. 2022, 2022.

[19] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 922–929.

[20] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.

[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 and Data Mining*, 2021, pp. 364–373.

[22] R. Z. et al., "Spatial–temporal dynamic semantic graph neural network," *Neural Computing and Applications*, vol. 34, pp. 16 655–16 668, 2022.

[23] J. Wang, L. Liu, and M. Zhang, "Novel hybrid model for traffic flow prediction," in *Proceedings of the 15th International Conference on Intelligent Transportation Systems and Technologies*, 2023.

[24] H. Y. et al., “Deep multi-view spatial-temporal network for taxi demand prediction,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
[25] C. Xu, A. Zhang, C. Xu, and Y. Chen, “Traffic speed prediction: Spatio-temporal convolution network based on long-term, short-term and spatial features,” *Applied Intelligence*, vol. 52, pp. 2224–2242, 2022.
[26] X. Zhang, Y. Xu, Y. Shao, “Forecasting traffic flow with spatial–temporal convolutional graph attention networks,” in *Neural Computing and Applications Conference Proceedings*, vol. 34, pp. 15457–15479, 2022.
[27] M. Niepert, M. Ahmed, and K. Kutzkov, “Learning convolutional neural networks for graphs,” in *Proceedings of the 33rd International Conference on Machine Learning*, vol. 48, 2016, pp. 2014–2023.
[28] S. Wang, H. Dai, L. Bai, C. Liu, and J. Chen, “Temporal branching-graph neural ode without prior structure for traffic flow forecasting,” *Engineering Letters*, vol. 31, no. 4, pp. 534–1545.
[29] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph wavenet for deep spatial-temporal graph modeling,” *arXiv preprint arXiv:1906.00121*, 2019.
[30] Z. Tan, Y. Zhu, and B. Liu, “Learning spatial–temporal feature with graph product,” *Signal Processing*, vol. 210, 2023.
[31] K. X. et al., “Show, attend and tell: Neural image caption generation with visual attention,” in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015, pp. 2048–2057.
[32] P. V. et al., “Graph attention networks,” *arXiv preprint arXiv:1710.10903*, 2017.
[33] H. L. et al., “Detectornet: Transformer-enhanced spatial temporal graph neural network for traffic prediction,” in *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, 2021, pp. 133–136.
[34] M. Zhang, W. Zhou, J. Huang, K. Huang, and X. Tang, “Self-attention based chebnet recurrent network for traffic forecasting,” in *Proceedings of the 2022 Chinese Intelligent Systems Conference*, 2022, pp. 300–309.
[35] H. Wu, J. Wu, J. Xu, J. Wang, and M. Long, “Flowformer: Linearizing transformers with conservation flows,” in *International Conference on Machine Learning*, 2022.
[36] B. M. Williams and L. A. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal arima process: theoretical basis and empirical results,” *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.
[37] J. D. Hamilton, *Time Series Analysis*. Princeton University Press, 2020.
[38] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in Neural Information Processing Systems*, 2014.
[39] M. Li and Z. Zhu, “Spatial-temporal fusion graph neural networks for traffic flow forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 4189–4196.
[40] X. Sun, F. Chen, Y. Wang, X. Lin, and W. Ma, “Short-term traffic flow prediction model based on a shared weight gate recurrent unit neural network,” *Physica A: Statistical Mechanics and its Applications*, vol. 618, 2023.
[41] Q. Lai and P. Chen, “Leisn: A long explicit–implicit spatio-temporal network for traffic flow forecasting,” *Expert Systems with Applications*, vol. 245, 2024.
[42] 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 *Proceedings of the 39th International Conference on Machine Learning, Proceedings of Machine Learning Research*, vol. 162, pp. 11906–11917, 2022. “‘