

Article

Spatiotemporal Dynamic Multi-Hop Network for Traffic Flow Forecasting

Wenguang Chai ^{1,†}, Qingfeng Luo ^{1,†} , Zhizhe Lin ^{2,†} , Jingwen Yan ³, Jinglin Zhou ^{4,*}  and Teng Zhou ^{2,5,*} 

¹ School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China; chaiwg@gdut.edu.cn (W.C.); 2112205337@mail2.gdut.edu.cn (Q.L.)

² School of Cyberspace Security, Hainan University, Haikou 570228, China; linzhizhe@hainanu.edu.cn

³ College of Engineering, Shantou University, Shantou 515063, China; jwyan@stu.edu.cn

⁴ School of Philosophy, Fudan University, Shanghai 200433, China

⁵ Yangtze Delta Region Institute, University of Electronic Science and Technology of China, Quzhou 324003, China

* Correspondence: jinglin_zhou@fudan.edu.cn (J.Z.); teng.zhou@hainanu.edu.cn (T.Z.)

† These authors contributed equally to this work.

Abstract: Accurate traffic flow forecasting is vital for intelligent transportation systems, especially with urbanization worsening traffic congestion, which affects daily life, economic growth, and the environment. Precise forecasts aid in managing and optimizing transportation systems, reducing congestion, and improving air quality by cutting emissions. However, predicting outcomes is difficult due to intricate spatial relationships, nonlinear temporal patterns, and the challenges associated with long-term forecasting. Current research often uses static graph structures, overlooking dynamic and long-range dependencies. To tackle these issues, we introduce the spatiotemporal dynamic multi-hop network (ST-DMN), a Seq2Seq framework. This model incorporates spatiotemporal convolutional blocks (ST-Blocks) with residual connections in the encoder to condense historical traffic data into a fixed-dimensional vector. A dynamic graph represents time-varying inter-segment relationships, and multi-hop operation in the encoder's spatial convolutional layer and the decoder's diffusion multi-hop graph convolutional gated recurrent units (DMGCGRUs) capture long-range dependencies. Experiments on two real-world datasets METR-LA and PEMS-BAY show that ST-DMN surpasses existing models in three metrics.

Keywords: intelligent transportation systems; sustainability; traffic flow forecasting; spatiotemporal dependency learning; graph structure learning



Citation: Chai, W.; Luo, Q.; Lin, Z.; Yan, J.; Zhou, J.; Zhou, T. Spatiotemporal Dynamic Multi-Hop Network for Traffic Flow Forecasting. *Sustainability* **2024**, *16*, 5860. <https://doi.org/10.3390/su16145860>

Academic Editor: Elzbieta Macioszek

Received: 24 May 2024

Revised: 3 July 2024

Accepted: 8 July 2024

Published: 9 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As urban vehicle populations increase and traffic data proliferate, the advancement of intelligent transportation systems (ITS) is becoming increasingly crucial [1,2]. Predicting traffic flow accurately is vital for these systems, as it involves analyzing historical traffic data to forecast current and future road conditions [3]. Achieving these goals is essential for improving traffic management, reducing congestion, and enhancing the overall efficiency of ITS, which directly contributes to the improvement of human well-being [4,5]. Optimized traffic management reduces commute times and driver stress [6]. Additionally, enhanced public transportation systems promote a healthier, more sustainable lifestyle for urban residents by improving efficiency and reliability [7,8].

Traffic flow is influenced by both time and the interconnections of roadways, resulting in a complex network structure [9,10]. Hence, forecasting real-time and accurate traffic flow is highly challenging because of the complex spatiotemporal dependencies inherent in traffic flow. The challenges are twofold: Firstly, sophisticated spatial correlations are influenced by both nearby and distant nodes. Secondly, historical traffic data exhibit complex nonlinear relationships, complicating long-term forecasts. Long-term forecasts

(over 30 min) are crucial for traffic planning and management, helping predict and respond to events like congestion and accidents, while short-term forecasts (5–30 min) address immediate conditions.

Advancing industrial technology has enabled the widespread use of sensors and data collection devices in transportation networks, collecting large amounts of crucial data for research. Methods such as the historical average (HA), exponential smoothing, and autoregressive integrated moving average (ARIMA) [11], which are commonly used in traditional traffic forecasting, rely on the assumption of stationarity. However, this assumption is not typically applicable to the dynamic nature of traffic data. These methods frequently fall short of providing accurate traffic forecasts because they cannot capture the complex dynamic characteristics of traffic data.

Recently, deep learning traffic flow forecasting methods utilize various neural network architectures to model spatiotemporal features [12,13]. Inspired by sequential learning, some neural networks simulate the temporal impact of traffic fluctuations [14,15]. Convolutional neural networks (CNNs) are used to capture spatial relationships between adjacent regions [16], while recurrent neural networks (RNNs) address the temporal aspects [17]. SHARE [18] predicts parking space availability in cities by employing a hierarchical graph convolutional structure to model spatial autocorrelation between parking lots and an RNN to capture dynamic temporal dependencies. Additionally, PredRNN [19] is a new recurrent neural network that forms a unified representation of complex environments with decoupled memory units and uses zigzag memory flow to transfer information between layers, effectively capturing the dynamics of spatiotemporal data.

Graph neural networks (GNNs) are widely employed in research to capture spatiotemporal dependencies among variables [20,21]. Early approaches, such as those by [22,23], utilized predefined graph structures to represent dependencies, but often struggled to capture complex interconnections, resulting in inaccuracies. Park et al. [24] employed a self-attention mechanism for dynamic correlation computation between sensor pairs, but scaling to large-scale graphs is hindered by quadratic computational complexity. Wang et al. [25] proposed D-TGNM, a dynamic temporal graph neural network designed to predict urban traffic flow robustly even with incomplete data. This model incorporates Traffic BERT to capture dynamic spatial relationships in road networks, alongside a temporal graph neural network (TGNM) that analyzes traffic flow patterns while accounting for missing data. Despite the progress in GNNs, current methods encounter two primary challenges.

First, current studies often use static graph models to describe spatial correlations in traffic networks, which limits traditional traffic prediction methods to direct relationships between adjacent road segments [26–28]. However, actual traffic systems are influenced by complex spatial interdependencies due to external factors such as weather changes, incidents, regulations, and temporal dynamics. Even geographically distant road segments may exhibit similar traffic fluctuation patterns over time. Focusing only on local spatial interdependencies may overlook information from remote segments, leading to increased forecast uncertainty. Figure 1 depicts the alterations in relationships among road networks, nodes, and vehicles influenced by traffic control measures. To better model dynamic interactions between road segments, it is crucial to incorporate dynamic graph models. Existing graph neural networks (GNNs) typically propagate information from immediate neighbors during message passing, thereby constraining their capability to capture long-range spatial dependencies [29].

Secondly, current neural network architectures face difficulties in modeling multi-step predictions for non-stationary time series data. Traffic data often show autocorrelation at adjacent time points and periodic fluctuations due to human activities, which vary in pattern and intensity across different periods and weekdays [30]. Researchers have explored various strategies to handle these complex temporal dependencies in multi-step forecasting. One approach involves temporal convolutional networks (TCNs) [31,32], which use convolution operations along the temporal dimension to process time series data efficiently, avoiding issues like gradient vanishing or exploding. While TCNs perform

well at single-step predictions with parallel processing, they may not be as effective for multi-step forecasting scenarios. Another method uses sequence-to-sequence models based on recurrent neural networks (RNN) (Seq2Seq) [22,33], which are adept at capturing the sequential nature of time series data. However, Seq2Seq models require multiple iterations to calculate long-distance dependencies [34], and their training phase lacks parallel computing, which could affect computational efficiency and graph-related structural information in pre-processing [35]. Hybrid methods that combine time convolution with recurrent structures may capture time dependencies more accurately and efficiently. Further research is required to confirm the efficacy of such hybrid architectures.

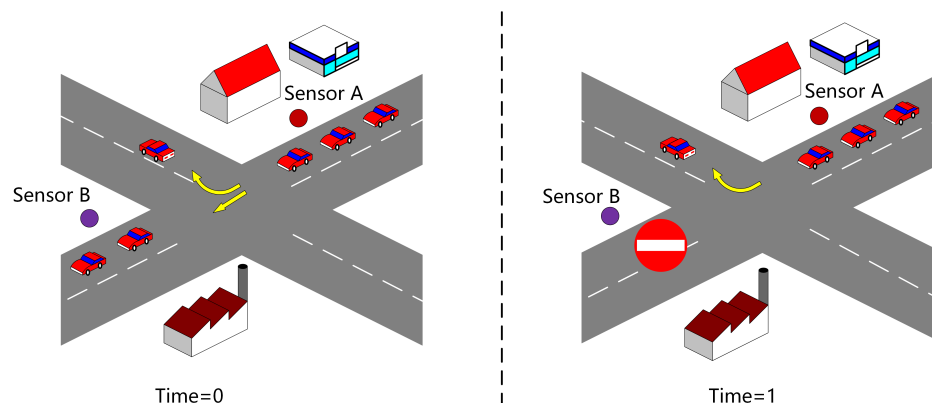


Figure 1. The relationship between the road grid, nodes, and vehicles. At time = 0, sensors A and B exhibit a strong correlation due to road connectivity; at time = 1, this correlation weakens as a result of traffic control measures.

This paper introduces a novel model architecture, named the spatiotemporal dynamic multi-hop network (ST-DMN), designed to tackle the aforementioned challenges. At the core of our approach is a dynamic graph learning algorithm tailored for each time step, adept at updating the graph structure in real time to reflect evolving temporal relationships. To address long-distance spatial dependencies, we have integrated multi-hop operation into our framework. Additionally, our model utilizes a transformer layer to introduce position coding to the spatiotemporal sequence generated by ST-Blocks. This approach not only improves the model's ability to capture the global temporal structure of the data but also emphasizes the interaction of relative positions within the sequence. In the domain of temporal correlation modeling, the ST-DMN employs an encoder–decoder architecture. The encoder in this architecture consists of L ST-Blocks designed to extract spatiotemporal features from historical traffic data. Subsequently, the decoder utilizes multiple diffusion multi-hop graph convolutional gated recurrent units (DMGCGRUs) as its foundational elements. These units enable the model to achieve multi-step prediction by performing autoregressive decoding of the extracted features. This research offers several key contributions:

- We have created an advanced encoder–decoder architecture tailored for multi-step prediction. The encoder extracts important spatiotemporal features from historical data by using multiple ST-Blocks. The decoder then uses DMGCGRUs to decode these features and produce multi-step prediction results.
- We present a dynamic graph learning algorithm to better capture the complex and evolving topology of traffic networks. This algorithm utilizes an iterative updating mechanism to dynamically construct and adjust the topological graph of road networks.
- ST-DMN combines the multi-hop operation of dynamic graphs with the diffusion convolution technique to effectively capture the inherent long-distance spatial dependence in traffic data. Furthermore, the transformer layer enhances the model's comprehension and perception of the overall temporal structure within the spatiotemporal embeddings generated by the ST-Blocks.

- Experimental results on publicly available traffic speed datasets, including METR-LA and PEMS-BAY, indicate that ST-DMN achieves competitive or even superior performance compared to various baseline models.

The structure of the paper is as follows: Section 2 provides a review of existing research and graph structure learning methods in the domain of traffic prediction. In Section 3, we provide a detailed explanation of the architecture of ST-DMN and its underlying algorithms. Sections 4 and 5 present a comprehensive comparison of the prediction accuracy of the ST-DMN model against several baseline models. It evaluates the influence of different parameters on the model's outcomes and compares training times, thereby demonstrating its superior performance. Section 6 concludes with the findings and discusses avenues for future research.

2. Related Works

2.1. Traffic Flow Forecasting

Pioneering techniques for anticipating traffic dynamics, including the historical average (HA), autoregressive integrated moving average (ARIMA) [11], and vector autoregression (VAR) [36], were founded on principles of mathematical statistics and rudimentary machine learning algorithms. However, these models frequently faced challenges in addressing the complexity and dynamics of real-world traffic conditions. Traditional traffic research has often oversimplified forecasting by treating it primarily as a time series problem, overlooking the substantial impact of other network nodes on traffic conditions. To improve predictive performance, researchers have explored innovative methods to capture the spatial characteristics of traffic networks. For example, ConvLSTM [37] uses convolutional operations to capture spatial features within temporal sequences. Graph convolutional neural networks (GCNs) extend traditional convolution concepts to non-Euclidean data structures. Spatiotemporal graph neural networks (STGNNs), such as STGCN [23], integrate graph convolution with temporal convolution or recurrent neural networks to adeptly capture spatiotemporal features within road networks.

In recent years, advanced models like GraphWaveNet [31] and AGCRN [38] have gained attention for their ability to dynamically learn the graph structure of road networks from traffic data. These models achieve predictive performance comparable to models based on predefined graphs. Deep neural networks, known for their powerful predictive capabilities, have been integrated with graph neural networks (GNNs) to construct spatiotemporal graph neural networks, enhancing prediction performance [22,39]. These models can be categorized by temporal modeling approaches into RNN-based, CNN-based, and attention-based models. For example, DCRNN [22] combines diffusion graph convolution with GRU for traffic forecasting, and STGCN [23] captures spatiotemporal correlations using 1D CNN and graph convolution. Transformer-based models like STAEFormer [40] and HTVGNN [41] have made significant strides in predictive accuracy and performance by introducing adaptive embeddings and time-varying graph structures.

2.2. Graph Structure Learning

Graph structure learning is a crucial field in machine learning that aims to extract structural information from data. In traffic forecasting, the graph structure models the road network, with nodes representing road intersections or segments and edges indicating traffic capacity or distance relationships. One of the primary challenges in learning graph structures is effectively capturing the intricate interdependencies between nodes.

Lately, academic research has been concentrated on improving the performance of graph neural networks (GNN) by optimizing graph structures [42–44]. Current graph structure learning methods can generally be categorized into three main groups: (1) Metric learning strategies, which assess inter-node relationships using various metrics [45,46]. (2) Probabilistic modeling approaches involve constructing graphs by sampling from probability distributions and modeling edge generation probabilities using trainable parameters [47,48]. (3) Direct optimization techniques optimize the entire graph alongside

GNN parameters, as explored in recent studies [49]. In traffic forecasting, recent studies have aimed to learn adaptable graph adjacency matrices [31]. However, many methods directly engineer trainable parameters for graph structures, often overlooking node attributes. This oversight can complicate model optimization, especially in cases with limited and sparse training data. Node attributes that provide details on traffic conditions are crucial for constructing a more accurate graph structure.

Recent academic studies, such as [33,44,50], have been concentrating on developing methods to accurately uncover dependencies between nodes. Advanced techniques involve extracting features from large traffic datasets to create graphs, which are then improved with prediction models to enhance prediction performance. As demonstrated by [51], this structured learning approach significantly enhances prediction accuracy. Our proposed structured learning framework goes beyond the analysis of current traffic conditions to deeply investigate traffic patterns. This comprehensive approach ensures the universality and suitability of the framework across different cities and regions, thereby improving its accuracy and scalability.

3. Materials and Methods

3.1. Problem Formulation

Traffic flow forecasting aims to accurately predict future traffic flow based on historical data. To accomplish this, we represent the road network as a weighted directed graph $\mathcal{G} = (V, E, W)$. Here, V denotes the set of nodes, e.g., sensors used to collect traffic flow data, with $|V| = N$. E represents the road segments, and $W \in \mathbb{R}^{N \times N}$ is a weighted adjacency matrix representing the proximity of a node. The element $W_{i,j}$ signifies the distance between nodes, typically determined by the road network or topological adjacency.

$$W_{i,j} = \exp\left(-\frac{\text{dist}(v_i, v_j)^2}{\sigma^2}\right) \quad \text{if } \text{dist}(v_i, v_j) \leq \kappa, \text{ otherwise } 0, \quad (1)$$

let $W_{i,j}$ indicate the weight of the edge connecting sensor v_i to sensor v_j , and $\text{dist}(v_i, v_j)$ represent the distance between sensor v_i and sensor v_j within the road network. Here, σ stands for the standard deviation of the distances, and κ is the designated threshold.

The traffic flow on \mathcal{G} is represented by a graph signal $X \in \mathbb{R}^{N \times D}$, where D is the number of features of each node, i.e., speed and flow rate. Traffic flow characteristics encompass various attributes, including traffic flow rate, average speed, and flow density. In this paper, the feature used is speed. The task of predicting traffic data for the next Q time intervals, given the historical data from the previous P time intervals, can be formulated as follows:

$$[X_{t-P+1}, \dots, X_t; \mathcal{G}] \xrightarrow{f_\theta} [\hat{Y}_{t+1}, \dots, \hat{Y}_{t+Q}], \quad (2)$$

where θ denotes the learnable model parameters.

3.2. Model Architecture

In this section, we introduce our spatiotemporal dynamic multi-hop network (ST-DMN), which is designed to capture spatiotemporal dependencies in traffic data. Our model employs an encoder–decoder architecture for multi-step traffic flow forecasting, as illustrated in Figure 2.

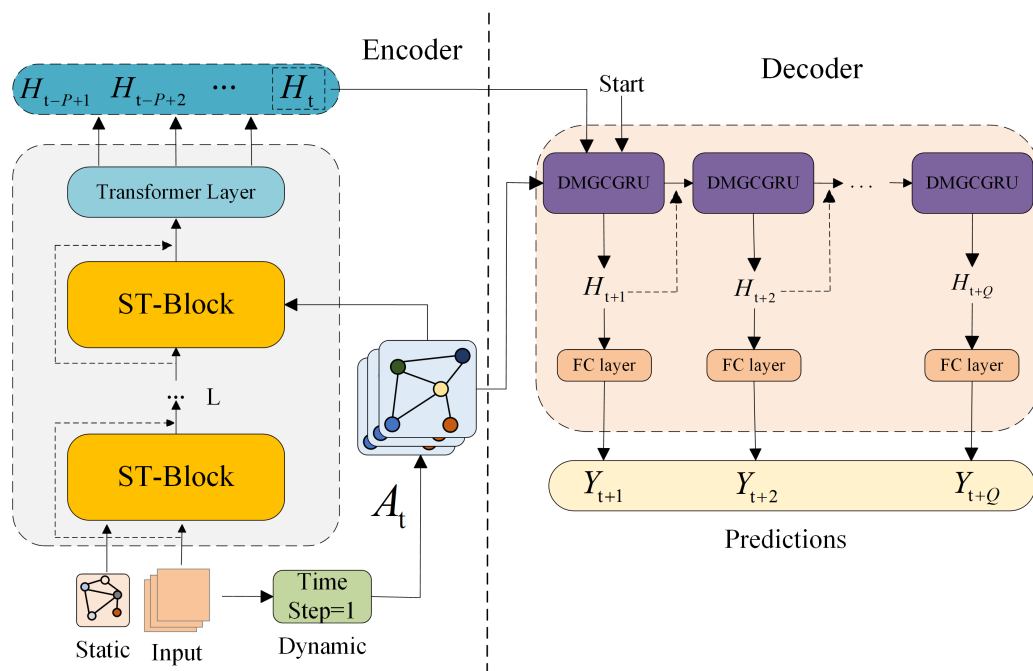


Figure 2. The framework of the spatiotemporal dynamic multi-hop network (ST-DMN).

The encoder comprises L spatiotemporal blocks (ST-Blocks). Each ST-Block is composed of two temporal convolution layers and one spatial convolution layer. The temporal convolution layer captures temporal features, while the spatial convolution layer extracts spatial features. We have integrated residual connections [52] to enhance information flow and address the vanishing gradient issue. Subsequently, the spatiotemporal embedding extracted by the L ST-Blocks is fed to a transformer layer, which enables a better understanding and perception of the global temporal structure of spatiotemporal features.

The decoder is made up of Q diffusion multi-hop graph convolutional gated recurrent units (DMGCCGRUs). A DMGCCGRU is an improved version of the gated recurrent unit (GRU) that replaces the traditional matrix multiplication in GRUs with diffusion convolution. This modification enhances the efficiency of managing spatiotemporal information transfer. By integrating multi-hop dynamic graph convolution, the DMGCCGRUs can better understand relationships across different locations and times. The first DMGCCGRU in the sequence takes the feature vector encoded by the encoder, ensuring that the spatiotemporal features captured during encoding smoothly transition into the decoding phase. The decoder interprets this vector to reconstruct the output. At each time step i within the interval $[t + 1, \dots, t + Q]$, the DMGCCGRUs receive the context vector generated from the previous step and produce a prediction using diffusion multi-hop convolution. Integrating diffusion convolution substantially improves the model's capacity to capture long-term spatiotemporal dependencies. During the training phase, we employ scheduled sampling [53] to alleviate the error accumulation that can arise from multi-step output predictions.

3.3. Encoder Architecture

3.3.1. ST-Blocks

The ST-Blocks integrate temporal convolution and spatial convolutions on graph-structured data to extract intricate dynamic spatial and temporal patterns in traffic data, as shown in Figure 3. Each ST-Block includes two temporal convolution layers and one spatial convolution layer. The initial temporal convolution layer uses convolutional kernels that slide along the temporal dimension to extract temporal features of the input traffic data. Subsequently, the spatial convolution layer learns the spatial dependencies by utilizing spatial convolution operations on the weighted adjacency matrix and performing multi-hop operations on the dynamic adjacency matrix to infer the spatial relationships between

nodes. After the spatial convolution layer, the *ReLU* activation function is applied to enhance the neural network's nonlinear modeling capability. Further, the second temporal convolution layer strengthens the time-varying relationship across the road segments and extracts hidden temporal features. A residual connection is implemented to capture and transmit important information, thereby avoiding the issue of gradient disappearance in deep networks. Finally, layer normalization is used to ensure the stability of the network training process.

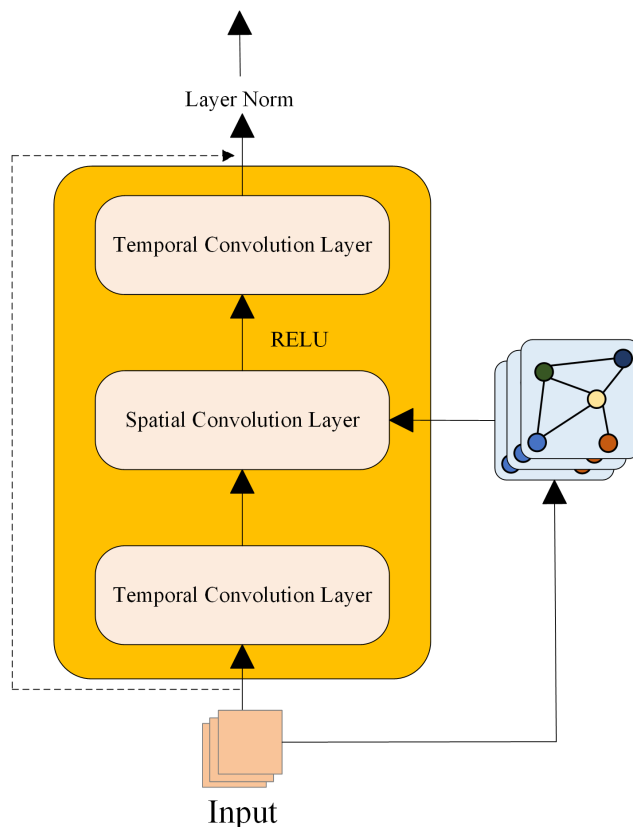


Figure 3. The architecture of ST-Block consists of two temporal convolutional layers and one spatial convolutional layer, with the utilization of residual connections for the output.

ST-Blocks integrate temporal and spatial convolutions, making them well-suited for complex spatiotemporal tasks that require extracting intricate spatiotemporal dependencies, and they exhibit strong training stability and generalization ability.

3.3.2. Dynamic Graph Learning

In a real traffic network, the interdependence among road segments varies and continuously evolves due to changing conditions such as time and weather. Traditional traffic forecasting methods often assume fixed interdependence, ignoring dynamic characteristics, and limiting prediction accuracy. To overcome this limitation, the study introduces a new method that no longer relies solely on a static adjacency matrix to represent the relationship between nodes. We incorporate the dynamically updated adjacency matrix into the model, as shown in Figure 4.

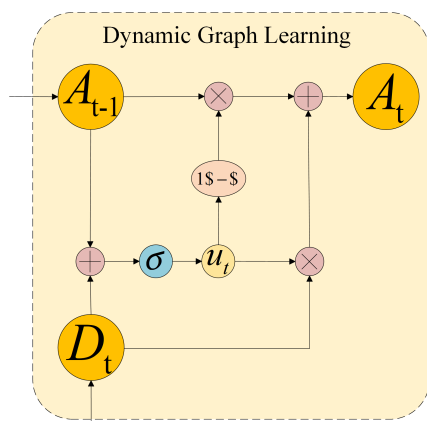


Figure 4. The process of dynamic graph learning demonstrates how the dynamic graph at the current time step can be generated from the dynamic graph of the previous time step through the use of an update gate.

This matrix primarily includes the dynamic graph from the previous time step, the adjacency matrix at the current time step, and the dynamic graph at the current time step. The dynamic graph learning process will be detailed below. This enhancement aims to enhance the accuracy and applicability of traffic flow forecasting, making it more adaptable to changes in traffic flow data.

Our proposed algorithm takes into account the nonlinear changes in adjacency relationships over time, which are observed in real traffic networks. It involves using a time-varying adjacency matrix and node embeddings to capture this dynamic interaction. Distinct from existing studies, our approach utilizes a three-dimensional tensor $D_t \in \mathbb{R}^{T \times N \times N}$ to represent the temporal changes in adjacency relations. Here T denotes the time steps and N denotes the number of nodes in the dataset. By embedding the time series X through a specific layer, we obtain graph signal embeddings, denoted as $X_t^{(1)}$ and $X_t^{(2)}$. This facilitates the learning of dynamic interactions among traffic network nodes at individual time steps. The specific learning process is carried out using the following dynamic spatial dependency computation formula:

$$D_t = SoftMax\left(ReLU\left(X_t^{(1)} \cdot (X_t^{(2)})^\top\right)\right), \tag{3}$$

$X_t^{(1)}$ and $X_t^{(2)}$ represent embeddings of one-hop neighboring nodes, and $D_t \in \mathbb{R}^{T \times N \times N}$ represents the adjacency matrix at time t .

During this computational process, the model accurately captures and updates the strength of interactions between nodes at different time slots. It is important to note that the adjacency matrix of a node obtained through this method represents the embedding of graph signals at the current time slot. As a result, the adjacency relationships can be dynamically adjusted based on optimization objectives.

In addition, the study proposes an iterative updating mechanism to preserve spatial relationships across consecutive time intervals. This mechanism continuously improves the adjacency matrix over successive time steps, allowing the model to not only represent the current traffic data but also smoothly transition to the next moment. The approach ensures that the model fully considers continuity within the temporal dimension. Thus, at each time step t , the core updating process of the dynamic graph updating algorithm can be represented by the following steps:

$$\begin{aligned} A_0 &= D_0, \\ U_t &= \sigma(\theta \odot (D_t + A_{t-1})), \end{aligned} \tag{4}$$

initialize dynamic the adjacency matrix A and set it to the initial state of D at time step 0. Compute the update gate U_t , which is a parameterized transformation activated by the *Sigmoid* activation function. It combines the information from the current time step D_t and the prior dynamic adjacency matrix state A_{t-1} . \odot denotes the hadamard product, and θ represents the learnable parameters. Adjust the state of the dynamic adjacency matrix A at time step t by employing the update gate U_t to maintain a balance between the stability and the dynamism of the dynamic graph:

$$A_t = U_t \odot D_t + (1 - U_t) \odot A_{t-1}, \quad (5)$$

Through this approach, the model proposed in this study not only excels in capturing time-varying traffic characteristics but also demonstrates outstanding scalability and robustness in long-term time series forecast tasks.

3.3.3. Temporal Convolution Layer

When analyzing and forecasting spatiotemporal data, it is crucial to consider the temporal dependency within time series data. In time series analysis, RNN-based methods are commonly used but often encounter challenges such as high computational complexity and issues with exploding or vanishing gradients, especially when dealing with long-range dependencies in real-world applications. To tackle these challenges, this study proposes a strategy that involves stacking temporal convolutional layers within the encoder. This approach improves the model's sensitivity and its capability to model spatiotemporal information. The temporal convolutional layers effectively capture and model the temporal dependencies within the input data sequences through convolution operations and gated linear units (GLU) [54].

In the traffic network \mathcal{G} , each node undergoes processing by the temporal convolutional layer, where padding is applied to keep the time step length P unchanged following the convolution operation. This helps preserve the temporal sequence of the data. Given an input tensor $X \in \mathbb{R}^{B \times P \times N \times D}$, the temporal convolutional layer uses convolutional kernels of size k_t , with an input dimension of D and an output dimension of $2D$. These kernels execute temporal convolution operations, efficiently mapping the input data features into a space of higher dimensionality. The convolution operation $*$ is conducted across multiple channels, with each channel undergoing independent convolution. The outcomes of all channels are then aggregated to produce an output with the dimension of $2D$. This output is subsequently divided into two equal parts along the feature dimension and is subsequently input into the *GLU*. The mathematical expression for the temporal convolutional layer is formulated as:

$$C = W * X + b, \quad (6)$$

where C represents the output of dimension $2D$ after convolution, W denotes the convolutional kernel weights, and b stands for the bias term.

$$GLU(X) = \sigma(C_1) \odot C_2, \quad (7)$$

$$T(X) = (GLU(C) + X) \cdot \theta, \quad (8)$$

where σ represents the activation function *Sigmoid* applied to the first half channels of C , while C_2 denotes the last half channels of C . The symbol \odot indicates element-wise multiplication. The output of the temporal convolutional layer is scaled by θ to enhance stability. Additionally, the gated linear unit (GLU) is utilized for non-linear transformation, aiding in accelerating model training and improving generalization capability.

3.3.4. Spatial Convolution Layer

The dynamics of traffic conditions can vary significantly across different roads and change constantly. To effectively capture long-distance spatial relationships, we have decided to move away from using the traditional multilayer graph neural network (GNN)

model. Previous research [55] has pointed out that traditional multi-layer GNNs suffer from over-smoothing, where node representations in locally connected subgraphs become overly similar, resulting in decreased prediction accuracy. We have developed an innovative spatial convolutional layer to adaptively capture the interrelationships between sensors in the road network, addressing this challenge effectively. We use a single-layer approach in this study to update node representations and integrate information from multi-hop neighbors. This method not only comprehensively captures the traffic topology of the road network, but also effectively mitigates the over-smoothing phenomenon. As part of this innovative approach, we have constructed a multi-hop graph model to operationalize our methodology:

$$A^{(k+1)} = (1 - \alpha)\alpha\tilde{A} + (1 - \alpha)A^{(k)}\tilde{A}, \quad (9)$$

where $\alpha \in (0, 1)$ is the attenuation factor used to control the degree of weight attenuation transmitted through the multi-hop path. \tilde{A} is the one-hop matrix, which is the dynamic adjacency matrix, and k is the number of hops.

In addition to learnable dynamic graphs, predefined adjacency matrices are also crucial. The matrix represents the fixed links among various parts of the urban network. Spatial proximity information is usually provided by the dataset itself or can be accessed through city maps. We model spatial dependencies by correlating time series to diffusion processes, explicitly capturing the random nature of traffic dynamics. An explicit graph S represents these correlations, with the diffusion process characterized by random walks on graph \mathcal{G} .

One limitation of basic graph convolution is its restriction to undirected graphs, which does not suit the directed nature of traffic networks. To address this, Li et al. [22] introduced forward and backward diffusion processes for graph signals with K finite steps, enabling convolution on directed graphs. Studies show that the bidirectional diffusion convolution technique successfully models both upstream and downstream traffic impacts in predictive models. The diffusion convolution operation is defined as follows:

$$S(X, W) = \sum_{k=0}^{K-1} \left(W_{k,1} \left(D_O^{-1} W \right)^k + W_{k,2} \left(D_I^{-1} W^\top \right)^k \right) X, \quad (10)$$

where $W_{k,1}$ and $W_{k,2}$ are the learnable filter parameters. K denotes the number of diffusion steps. $D_O^{-1} W$, $D_I^{-1} W^\top$ represent the transfer matrices for the diffusion process (output) and the reverse diffusion process (input), respectively.

In this study, we couple the static graph generated by prior relationships with the dynamic graph produced through multi-hop processing, ultimately proposing the spatial convolution layer of this paper, represented as follows:

$$Z_t^{(k+1)} = S(X, W) + D_{(t)}^{(k+1)}(X, \tilde{A}), \quad (11)$$

where $Z_t^{(k+1)}$ represents the output signal of the spatial convolution layer. $S(\cdot)$ and $D(\cdot)$ denote the outputs of the static and dynamic graphs after the diffusion convolution layer, respectively.

The spatial convolution layer uses both static graphs and dynamic multi-hop graphs to handle spatial relationships. It makes use of static graphs to address underlying spatial dependencies, while dynamic multi-hop graphs are particularly effective at identifying long-distance spatial connections. We have employed an integrated strategy that combines dynamic graph modeling with multi-hop operation. This fusion allows the network to flexibly map dynamic traffic topology changes and simulate interactions between different road network sections over various time points. This integrated approach ensures a comprehensive understanding of spatial relationships. It enhances the network's adaptability and responsiveness to fluctuations in traffic patterns.

3.3.5. Transformer Layer

Recently, there has been increasing interest in neural network architectures that effectively model the internal dependencies of spatiotemporal data. In this paper, we introduce a transformer layer module specifically designed to handle spatiotemporal information. Our module integrates a self-attention mechanism, a convolutional layer, and positional encoding to capture complex patterns in a flexible and scalable manner.

The positional encoding augments the sequence information of the spatiotemporal embedding, enabling the model to accurately distinguish information across different time steps and effectively manage long-term dependencies. Enhancing sensitivity to spatiotemporal relationships, the self-attention mechanism enables the model to dynamically focus on information from different locations when processing sequence data. Additionally, the convolution layer is used to extract local features, thereby enhancing the model's capability to perceive the local structure of the input data space. To further stabilize and expedite the model's convergence during training, particularly with long sequence data, we have introduced a layer normalization mechanism. This module uses as input the spatiotemporal embedding $H \in \mathbb{R}^{B \times T \times N \times D}$ processed by ST-Blocks, expressed as follows:

$$H_{(PE)} = H + PE(H), \quad (12)$$

$$\begin{aligned} Q &= Conv_1(H_{(PE)}), \\ K &= Conv_2(H_{(PE)}), \end{aligned} \quad (13)$$

$$V = Linear(H_{(PE)}),$$

$$A = SoftMax\left(\frac{QK^T}{\sqrt{C}}\right), \quad (14)$$

$$H_{out} = LayerNorm(AV + H_{(PE)}), \quad (15)$$

where PE is the matrix calculated based on the positional encoding, the input $H_{(PE)} \in \mathbb{R}^{B \times T \times N \times D}$ of the transformer block. $Conv_1$ and $Conv_2$ are convolution operations, and $Linear$ is a linear transformation. A denotes the self-attention matrix, $H_{out} \in \mathbb{R}^{B \times T \times N \times D}$ represents the output of this module.

3.4. Decoder Architecture

We utilize Q diffusion multi-hop graph convolutional gated recurrent units (DMGCGRUs) as decoders, as illustrated in Figure 5, to independently capture static and multi-hop spatial dependencies. Specifically, we replace the matrix multiplication of the traditional gated recurrent unit (GRU) with diffusion convolutions. This modification allows us to input the context vector at each time step t within the range $t = [t + 1, \dots, t + Q]$. Multi-hop dynamic graph convolution, \mathcal{G}_Z in Figure 5, is introduced in the process of dynamic spatial dependency computation, and this design broadens the receptive field of the target node and adaptively captures the fluctuations in the connectivity of the road network. The computational process for the DMGCGRUs at time t is as follows:

$$\begin{aligned} u_t &= Sigmoid(\mathcal{G}_Z(X_t \oplus H_{t-1})\theta_u + b_u), \\ r_t &= Sigmoid(\mathcal{G}_Z(X_t \oplus H_{t-1})\theta_r + b_r), \\ C_t &= Tanh(\mathcal{G}_Z(X_t \oplus (r_t \odot H_{t-1}))\theta_C + b_C), \\ H_t &= (1 - u_t) \odot C_t + u_t \odot H_{t-1}, \end{aligned} \quad (16)$$

among them, X_t and H_t denote the input signal and hidden state signal at time t , respectively. r_t and u_t denote the reset gate and update gate at time t , respectively. The symbol \odot denotes the Hadamard product. The vector concatenation is denoted by \oplus . \mathcal{G}_Z represents

the multi-hop dynamic graph convolution layer as given in Equation (11), with $\theta_u, \theta_r, \theta_C$ being the parameters of the corresponding filters.

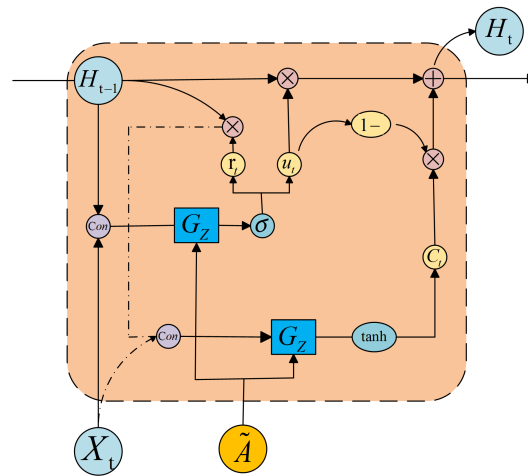


Figure 5. The architecture of DMGCCGRU. It contains dynamic graph multi-hop operations and static graph coupling processes \mathcal{G}_Z and GRU units.

3.5. Loss Function

In this study, the loss function used is the mean absolute error (MAE), formulated as shown in Equation (17).

$$\mathcal{L}(\theta) = \frac{1}{T} \frac{1}{N} \sum_{t=1}^T \sum_{i=1}^N |Y_{i,t} - \hat{Y}_{i,t}(\theta)| \quad (17)$$

where θ denotes all trainable parameters, $\hat{Y}_{i,t}$ denotes the predicted output, and $Y_{i,t}$ represents the ground truth of the i -th node at time t .

4. Experiments

4.1. Datasets

Experiments are performed on two real-world large-scale traffic datasets: METR-LA and PEMS-BAY. METR-LA consists of 207 nodes, while PEMS-BAY comprises 325 nodes. Specific statistics can be found in Table 1.

To ensure a fair comparison with other benchmark methods, we adhered to the data processing procedure outlined in the paper [22]. The dataset was split chronologically, with 70% used for training, 20% for testing, and the remaining 10% for validation. Furthermore, all input speed data were normalized using the Z-score method [22,31,56].

Table 1. Description and statistics of METR-LA and PEMS-BAY datasets.

Dataset	Nodes	Edges	Time Steps	Data Points
METR-LA	207	1515	34,272	6,519,002
PEMS-BAY	325	2369	52,116	16,937,179

4.2. Baselines

To assess the effectiveness of ST-DMN, we benchmarked it against various traditional and advanced model methodologies. Our focus was on the prediction errors associated with the 15, 30, and 60 min forecasts. Here is a refined overview of the comparative baselines we considered:

- HA: employs the mean of past data as a basis for forecasting subsequent traffic volumes.

- ARIMA [57]: combines autoregression, differencing, and moving average to forecast non-stationary time series.
- FC-LSTM [58]: combines fully connected layers and LSTM for enhanced time series prediction.
- DCRNN [22]: it is a deep learning model that integrates a graph convolutional network with the recurrent neural network and enhances the model's understanding of spatial correlation by a diffusion convolution operation.
- Graph WaveNet [31]: uses graph convolutional networks and dilated convolutions to capture spatiotemporal traffic patterns.
- GMAN [59]: adopts an encoder–decoder architecture with spatiotemporal attention blocks for dynamic traffic prediction.
- CCRNN [60]: integrates spatial and temporal features using coupled graph convolutions and gated recurrent units.
- GTS [33]: learns probabilistic graph structures for multiple time series prediction.
- PM-MemNet [61]: uses a memory network with pattern matching to predict traffic in complex road networks.

4.3. Experiment Settings

All deep learning-based models, including our ST-DMN, were implemented using Python 3.8 and PyTorch 1.10.0 and executed on a GPU server equipped with one RTX 3090/24 GB. We conducted 5 experiments for each model, and the final results were averaged to ensure reliability and statistical significance.

During our experiments, we optimized all hyperparameters by minimizing the MAE metric. According to previous work [22], the number of input steps equals the number of output steps. For instance, we used $P = 12$ steps (equivalent to 60 min) to predict the next $Q = 12$ steps (60 min). The model included $L = 2$ ST-Blocks, allowed for a maximum diffusion of a $K = 1$ step, used an embedding dimension of DGL equal to 16, and employed a batch size of 64. During training, the Adam optimizer [62] was used with a fixed learning rate of 0.01, decayed by 0.5 every 10 epoch. Training sessions lasted for 100 epochs, with early stopping implemented to prevent overfitting.

4.4. Evaluation Metrics

Suppose Y denotes the ground truth, \hat{Y} denotes the predicted values, and N denotes the total number of samples. The evaluation metrics used in this study include: mean absolute error (MAE), measured in mph, root mean square error (RMSE), also measured in mph, and mean absolute percentage error (MAPE), measured in %. They are widely used as an evaluation index in traffic state prediction tasks.

$$MAE = \frac{1}{N} \sum_{i=1}^N |Y - \hat{Y}| \quad (18)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y - \hat{Y})^2} \quad (19)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{Y - \hat{Y}}{Y} \right| \quad (20)$$

5. Results

5.1. Experiment Results and Analysis

The results presented in Table 2 compare the ST-DMN model with other baseline models based on predictions made 15 min (3 steps), 30 min (6 steps), and 60 min (12 steps) in advance using the METR-LA and PEMS-BAY datasets. Superior predictive performance is highlighted in bold, while suboptimal performance is underlined. Our model demon-

strates superior performance, indicated by bold highlighting. Based on these experimental findings, the following conclusions can be drawn:

- (1) Deep learning methods demonstrate superior performance over traditional time series methods and machine learning models. Traditional approaches often face challenges in achieving high data stationarity, whereas deep neural networks excel in modeling the nonlinear dynamics of traffic data.
- (2) Among deep learning models, graph-based architectures like DCRNN, GW-Net, and GMAN consistently outperform FC-LSTM models. This emphasizes the significance of integrating road network data into traffic flow forecasting models, indicating that spatial connectivity plays a critical role in accurate prediction.
- (3) The CCRNN model initializes its learnable graph using a 0–1 adjacency matrix of the road network, while the GTS model transforms the problem into learning a probabilistic graphical model by optimizing the performance averaged across the graph distribution. These models leverage dynamic graph structures to enhance predictive performance over earlier methods.
- (4) The PM-MemNet model innovatively uses a key-value memory structure to associate input data with representative patterns and identify the best pattern for predicting future traffic conditions based on given spatiotemporal features.

At present, the PM-MemNet model represents the forefront of traffic forecasting and shows superior performance.

Table 2. Comparison of forecasting performance between ST-DMN and other baseline models on the METR-LA and PEMS-BAY datasets.

Dataset	Models	15 min/Horizon 3			30 min/Horizon 6			60 min/Horizon 12		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
METR-LA	HA	4.16	7.80	13.00	4.16	7.80	13.00	4.16	7.80	13.00
	ARIMA [57]	3.99	8.12	9.60	5.15	10.45	12.7	6.90	13.23	17.40
	FC-LSTM [58]	3.44	6.30	9.60	3.77	7.23	10.90	4.37	8.69	13.20
	DCRNN [22]	2.77	5.38	7.30	3.15	6.45	8.80	3.60	7.60	10.50
	GW-Net [31]	2.69	<u>5.15</u>	6.90	3.07	<u>6.22</u>	<u>8.37</u>	3.53	7.37	10.01
	GMAN [59]	2.80	5.55	7.41	3.12	6.49	8.73	3.44	7.35	10.07
	CCRNN [60]	2.85	5.54	7.50	3.24	6.54	8.90	3.73	7.65	10.59
	GTS [33]	2.65	5.22	<u>6.83</u>	3.09	6.34	8.45	3.59	7.29	9.83
	PM-MemNet [61]	<u>2.65</u>	5.29	7.01	<u>3.03</u>	6.29	8.42	3.46	7.56	10.26
ST-DMN	2.63	5.09	6.65	3.01	6.15	8.08	<u>3.45</u>	<u>7.32</u>	<u>9.91</u>	
PEMS-BAY	HA	2.88	5.59	6.80	2.88	5.59	6.80	2.88	5.59	6.80
	ARIMA [57]	1.62	3.30	3.50	2.33	4.76	5.40	3.38	6.50	8.30
	FC-LSTM [58]	2.05	4.19	4.80	2.20	4.55	5.20	2.37	4.96	5.70
	DCRNN [22]	1.38	2.95	2.90	1.74	3.97	3.90	2.07	4.74	4.90
	GW-Net [31]	<u>1.30</u>	<u>2.74</u>	2.70	<u>1.63</u>	3.70	<u>3.70</u>	1.95	4.52	4.60
	GMAN [59]	1.35	2.90	2.87	1.65	3.82	3.74	<u>1.92</u>	4.49	<u>4.52</u>
	CCRNN [60]	1.38	2.90	2.90	1.74	3.87	3.90	2.07	4.65	4.87
	GTS [33]	1.39	2.95	2.88	1.78	4.06	3.98	2.24	5.17	5.35
	PM-MemNet [61]	1.34	2.82	2.81	1.65	3.76	3.71	1.95	<u>4.49</u>	4.54
ST-DMN	1.30	2.74	<u>2.73</u>	1.62	<u>3.73</u>	3.67	1.89	4.46	4.51	

We compared the performance of the ST-DMN with the latest baseline model using the relative error rate metric. On the METR-LA dataset, our ST-DMN model showed improvements in the mean absolute percentage error (MAPE) of 5.41% for 3 steps, 4.21% for 6 steps, and 3.53% for 12 steps. When tested on the PEMS-BAY dataset, we observed improvements of 2.93%, 1.09%, and 0.67% for the 3, 6, and 12 steps ahead predictions, respectively. Overall, the experimental results demonstrate that ST-DMN performs strongly and competitively against both traditional and state-of-the-art baseline models. The ability of the model to leverage spatiotemporal information significantly enhances prediction

accuracy, making it a promising candidate for practical applications, including traffic management and urban planning.

Figure 6 compares the time series prediction performance of the four latest models on the METR-LA and PEMS-BAY datasets. As illustrated, the prediction errors for all models tend to increase as the forecasted time horizon extends. This trend is due to the increased uncertainty inherent in longer prediction spans. Overall, the ST-DMN model performs better than the others on both datasets, with suboptimal results for the one-hour forecast on METR-LA. This may be because the factors to be considered become more complex and uncertain as the prediction time increases, such as with accidents and weather conditions. The PM-MemNet model also demonstrates robust performance, particularly in short-term forecasting, likely owing to its efficient memory mechanism for handling time series data. Conversely, the CCRN model exhibits significant errors across all time domains, possibly due to its simpler structure struggling to capture the intricacies of spatiotemporal relationships.

In summary, the ST-DMN model shows more prominent performance on both datasets, likely due to its ability to effectively handle spatiotemporal dependencies in time series data. However, errors for all models increase as the forecast time interval lengthens, highlighting the inherent challenge of time series forecasting: longer forecasts come with greater uncertainty.

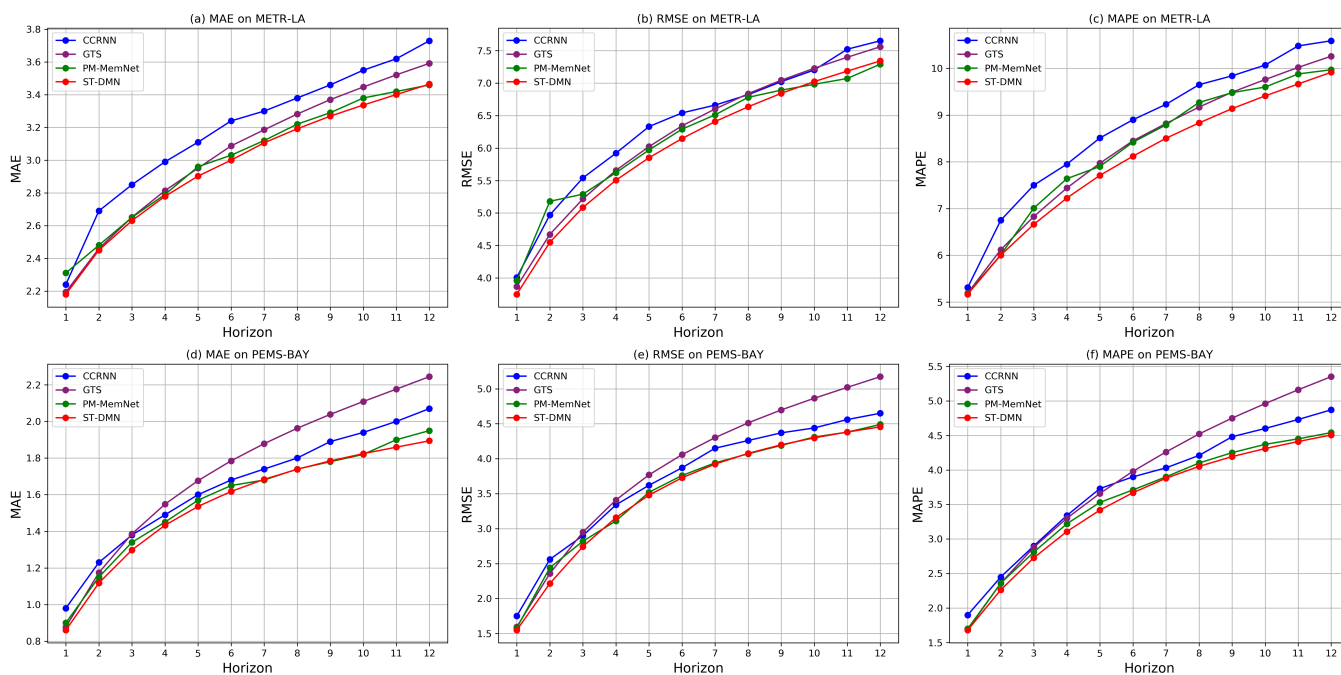


Figure 6. Prediction error at each horizon on METR-LA and PEMS-BAY.

5.2. Model Configuration Analysis

Throughout our study, we conducted several experiments to assess the impact of different parameter settings on model performance. Table 3 shows the average values of MAE, RMSE, and MAPE for various parameters.

We started by analyzing the parameter k in Equation (9). The experimental results were notably influenced by the chosen value of k . As k increases, the average values of the three metrics decrease, indicating an overall improvement in predictive performance. For instance, when $k = 3$, the MAE decreased by 0.04 compared to when $k = 1$, and the RMSE decreased by 0.1. This aligns with the findings in [63], demonstrating that by considering more distant neighbors through multi-hop paths, the model’s predictive outcomes are positively affected. However, we also noticed that when k reaches 4, the performance declined. This suggests that while a larger k captures a broader graph structure information,

it also increases computational complexity. Therefore, striking a balance between model performance and computational efficiency is crucial. Furthermore, it became apparent that the additional improvements in predictive performance become limited with further increases in k .

Next, we explored how the decay factor α in Equation (9) affects experimental outcomes. Our study revealed that a smaller α has a positive impact on the model, yielding the best results when $\alpha = 0.15$. The parameter α influences the amplification of low-frequency signals in the graph structure. A smaller α enhances low-frequency signals while suppressing high-frequency ones [63]. Selecting an appropriate α value allows the model to effectively capture graph structure information and improves prediction performance.

Lastly, we discussed the critical parameter of node embedding dimensions in DGL. Table 3 reveals that neither excessively large nor small embedding dimensions achieve optimal performance. Smaller embedding dimensions may limit the expressive power of node features, while larger dimensions increase computational complexity and risk overfitting. The model achieves the best results with an embedding dimension of 16, demonstrating its robustness.

Analyzing these parameters provides a better understanding of their impact on model performance, which facilitates more informed decisions in practical applications.

Table 3. Average performance metrics for different model configurations.

Configurations	k(hops)	α	Embedding	MAE	RMSE	MAPE (%)
k(hops)	1	0.15	16	3.01	6.12	8.02
	2	0.15	16	3.00	6.11	8.08
	3	0.15	16	2.97	6.02	7.98
	4	0.15	16	2.98	6.10	8.03
α	3	0.1	16	2.98	6.04	7.94
	3	0.15	16	2.97	6.02	7.98
	3	0.3	16	2.99	6.05	7.98
	3	0.4	16	2.99	6.07	7.99
	3	0.5	16	3.00	6.05	8.01
Embedding	3	0.15	8	3.00	6.12	7.99
	3	0.15	16	2.97	6.02	7.98
	3	0.15	32	3.01	6.10	7.99
	3	0.15	64	3.00	6.12	8.19

5.3. Model Efficiency

Table 4 presents a comparison of the computation times between ST-DMN and other state-of-the-art baseline models on the METR-LA dataset. We recorded the average training time per epoch for each model. One advantage of ST-DMN over DCRNN is that its encoder does not use a recurrent neural network, resulting in better computational performance. PM-MemNet utilizes graph convolutional memory networks (GCMem), which can increase computational costs when dealing with large-scale datasets. While GTS simplifies the optimization process with a single-stage optimization approach, the search for an optimal solution among many possible graph structures can increase computational demands. Despite the lack of a dedicated decoder, ST-DMN outperforms Graph WaveNet in predictive performance, although it is slightly slower in terms of computation speed. Therefore, for researchers looking for a balance between computational cost and performance, ST-DMN is an ideal choice, offering enhanced performance while maintaining high computational efficiency.

Table 4. Comparison of computation time of different models on the METR-LA dataset.

Models	ST-DMN	PM-MemNet	GTS	DCRNN	GW-Net
Time (s/epoch)	82.44	131.38	727.60	249.31	53.68

5.4. Ablation Study

To thoroughly validate our proposed model design, we conducted a series of ablation experiments to systematically analyze the impact of various components on performance. These experiments were evaluated using the METR-LA and PEMS-BAY datasets. The results, detailed in Table 5, offer a comprehensive overview of the model's performance across different prediction intervals: 15 min (3 steps), 30 min (6 steps), and 60 min (12 steps). The distinctive features of these variants are as follows:

- “w/o Transformer Layer” excludes the transformer layer.
- “w/o DGL” excludes the dynamic graph learning.
- “w/o Multi-Hop” excludes the multi-hop operation.

Table 5. The results of ablation experiments on METR-LA and PEMS-BAY datasets.

Dataset	Models	15 min/Horizon 3			30 min/Horizon 6			60 min/Horizon 12		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
METR-LA	w/o Transformer Layer	2.64	5.12	6.75	3.02	6.20	8.20	3.48	7.39	9.94
	w/o DGL	2.66	5.14	6.71	3.04	6.24	8.14	3.50	7.44	9.87
	w/o Multi-Hop	2.66	5.17	6.72	3.03	6.23	8.17	3.47	7.32	9.89
	ST-DMN	2.63	5.09	6.65	3.01	6.15	8.08	3.45	7.32	9.91
PEMS-BAY	w/o Transformer Layer	1.31	2.75	2.77	1.64	3.77	3.76	1.92	4.52	4.59
	w/o DGL	1.31	2.77	2.77	1.63	3.77	3.73	1.92	4.54	4.60
	w/o Multi-Hop	1.32	2.77	2.77	1.65	3.78	3.73	1.94	4.53	4.57
	ST-DMN	1.30	2.74	2.73	1.62	3.73	3.67	1.89	4.46	4.51

In our evaluation of the METR-LA dataset, we initially assessed the effects of excluding the transformer layer from the model. The results indicated a slight increase in MAE, RMSE, and MAPE across all prediction horizons for the model lacking the transformer layer. This finding emphasizes the critical role of the transformer layer in capturing spatiotemporal relationships within traffic data. Similarly, we investigated the performance impact of removing the dynamic graph learning (DGL) component. The results indicated a decrease in performance, highlighting the significance of dynamic graphs in learning spatial dependencies between traffic nodes. Furthermore, we examined the contribution of multi-hop operation to the model. The exclusion of multi-hop operation resulted in higher prediction errors, highlighting their essential role in modeling extended spatial dependencies. The model's ability to thoroughly capture traffic topology information within the road network and mitigate the issue of over-smoothing further highlights the importance of these operations.

The complete model, ST-DMN, incorporating all proposed components, demonstrates the best performance across all evaluation metrics and prediction periods of the METR-LA dataset. Similar trends as those on the METR-LA dataset were observed in the ablation experiments conducted on the PEMS-BAY. The performance decline of the ST-DMN model when excluding the transformer layer, DGL, and multi-hop operations are graphically represented in Figure 7. Consistently, the comprehensive ST-DMN model outperforms its counterparts lacking certain components across all evaluation metrics and different prediction horizons.

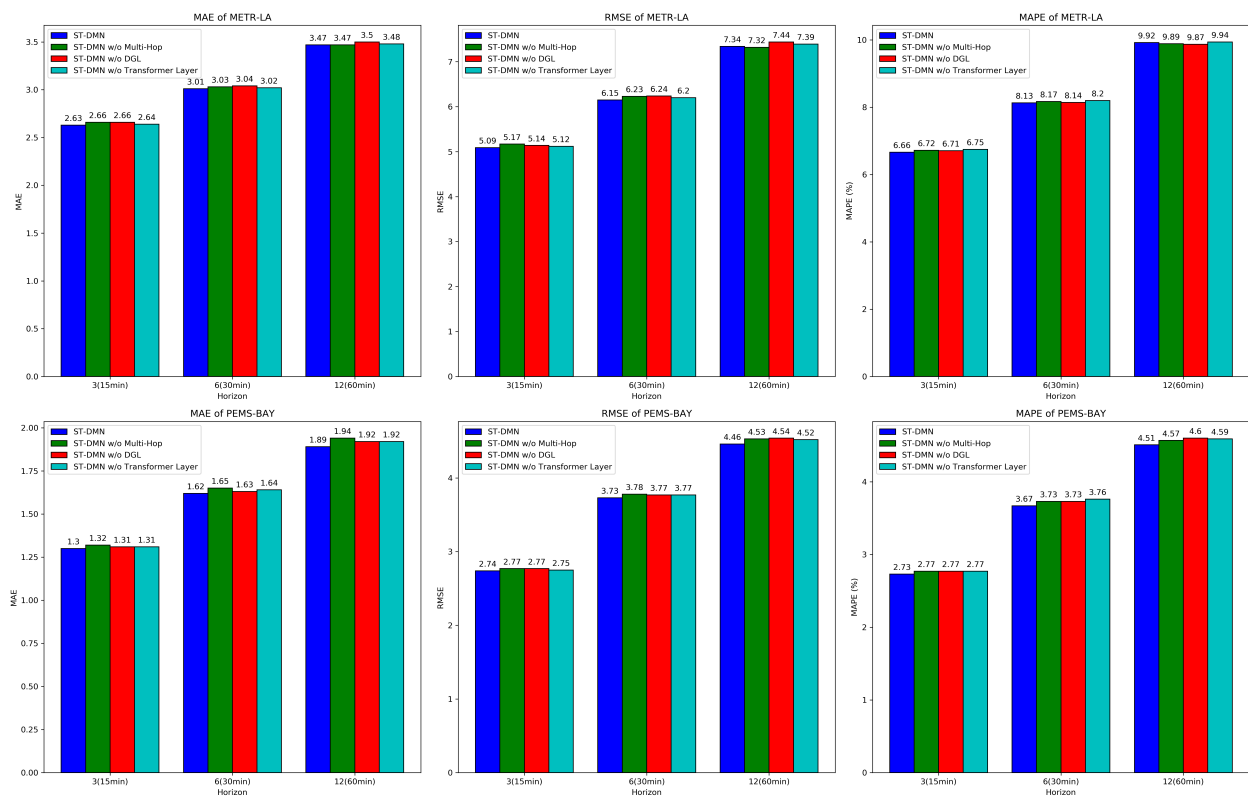


Figure 7. Comparison of ST-DMN variants on the METR-LA and PEMS-BAY dataset.

In summary, the ablation experiments highlight the effectiveness of the proposed components in the ST-DMN model and emphasize the synergistic effect of Transformer layers, DGL, and multi-hop operations in improving the overall prediction accuracy on the METR-LA and PEMS-BAY datasets. This detailed analysis further validates the importance of our proposed model architecture in capturing the spatiotemporal dependencies necessary for accurate traffic flow forecasting.

6. Conclusions

This study introduces the spatiotemporal dynamic multi-hop network (ST-DMN), an innovative architectural model designed to address the complexity of traffic flow forecasting. The core of ST-DMN is a dynamic graph learning algorithm that iteratively updates the traffic network graph to accurately represent the evolving traffic patterns. By integrating multi-hop operation with diffusion convolutional techniques, our model effectively captures long-distance spatial dependencies, an important aspect of traffic data that existing models fail to adequately address. In addition, the incorporation of transformer layers enables ST-DMN to effectively understand the global temporal structure, consequently enhancing its prediction ability. ST-DMN successfully captures the intricate spatiotemporal dependence of traffic speed and outperforms existing models in terms of prediction accuracy. In future work, we plan to explore the application of extended models, including validation on different cities or traffic network structures to evaluate its generalization performance.

Author Contributions: Conceptualization, W.C. and T.Z.; methodology, W.C. and Q.L.; software, Q.L. and Z.L.; validation, Z.L. and J.Y.; investigation, J.Y.; writing—original draft preparation, J.Z. and W.C.; supervision, J.Z. and T.Z.; project administration, T.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Guangdong Province (No. 2022A1515011590, 2024A1515011766), the China Postdoctoral Science Foundation (No. 2024M750562),

the National Postdoctoral Fellowship Program (No. GZC20230549), the State key laboratory major special projects of Jilin Province Science and Technology Development Plan (No. SKL202402024), the Natural Science Foundation of China (No. 82020108016, 61902232), the National Key Research and Development Program (No. 2021YFB2700600, 2022YFB3104600), the Guangdong Provincial University Innovation Team Project (No. 2020KCXTD012), the Municipal Government of Quzhou (No. 2022D018, 2022D029, 2023D007, 2023D015, 2023D033, 2023D034, 2023D035), and the Natural Science Foundation of Zhejiang Provincial (No. LGF22G010009).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset and source code generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors express their gratitude to the reviewers and editors for their valuable feedback and contributions to refining this manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Zhou, T.; Zhao, Y.; Lin, Z.; Zhou, J.; Li, H.; Wang, F. Moral and Formal Model-based Control Strategy for Autonomous Vehicles at Traffic-light-free intersections. *Smart Constr. Sustain. Cities* **2024**, *2*, 11. [\[CrossRef\]](#)
- Kaffash, S.; Nguyen, A.T.; Zhu, J. Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis. *Int. J. Prod. Econ.* **2021**, *231*, 107868. [\[CrossRef\]](#)
- Wang, F.; Liang, Y.; Lin, Z.; Zhou, J.; Zhou, T. SSA-ELM: A Hybrid Learning Model for Short-Term Traffic Flow Forecasting. *Mathematics* **2024**, *12*, 1895. [\[CrossRef\]](#)
- Voukelatou, V.; Gabrielli, L.; Miliou, I.; Cresci, S.; Sharma, R.; Tesconi, M.; Pappalardo, L. Measuring objective and subjective well-being: Dimensions and data sources. *Int. J. Data Sci. Anal.* **2021**, *11*, 279–309. [\[CrossRef\]](#)
- Park, C.L.; Kubzansky, L.D.; Chafouleas, S.M.; Davidson, R.J.; Keltner, D.; Parsafar, P.; Conwell, Y.; Martin, M.Y.; Hanmer, J.; Wang, K.H. Emotional well-being: What it is and why it matters. *Affect. Sci.* **2023**, *4*, 10–20. [\[CrossRef\]](#)
- Chui, K.T. Driver stress recognition for smart transportation: Applying multiobjective genetic algorithm for improving fuzzy c-means clustering with reduced time and model complexity. *Sustain. Comput. Inform. Syst.* **2022**, *35*, 100668. [\[CrossRef\]](#)
- Xu, W.; Liu, J.; Yan, J.; Yang, J.; Liu, H.; Zhou, T. Dynamic Spatiotemporal Graph Wavelet Network for Traffic Flow Prediction. *IEEE Internet Things J.* **2024**, *19*, 8019–8029. [\[CrossRef\]](#)
- Ghafouri-Azar, M.; Diamond, S.; Bowes, J.; Gholamalizadeh, E. The sustainable transport planning index: A tool for the sustainable implementation of public transportation. *Sustain. Dev.* **2023**, *31*, 2656–2677. [\[CrossRef\]](#)
- Li, Z.; Zhou, J.; Lin, Z.; Zhou, T. Dynamic spatial aware graph transformer for spatiotemporal traffic flow forecasting. *Knowl.-Based Syst.* **2024**, *297*, 111946. [\[CrossRef\]](#)
- Ishak, S.; Al-Deek, H. Performance evaluation of short-term time-series traffic prediction model. *J. Transp. Eng.* **2002**, *128*, 490–498. [\[CrossRef\]](#)
- Isufi, E.; Loukas, A.; Simonetto, A.; Leus, G. Autoregressive moving average graph filtering. *IEEE Trans. Signal Process.* **2016**, *65*, 274–288. [\[CrossRef\]](#)
- Wang, X.; Ma, Y.; Wang, Y.; Jin, W.; Wang, X.; Tang, J.; Jia, C.; Yu, J. Traffic flow prediction via spatial temporal graph neural network. In Proceedings of the WWW '20: The Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 1082–1092. [\[CrossRef\]](#)
- Shi, X.; Qi, H.; Shen, Y.; Wu, G.; Yin, B. A spatial-temporal attention approach for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4909–4918. [\[CrossRef\]](#)
- Liu, Q.; Wu, S.; Wang, L.; Tan, T. Predicting the next location: A recurrent model with spatial and temporal contexts. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30. [\[CrossRef\]](#)
- Yu, R.; Li, Y.; Shahabi, C.; Demiryurek, U.; Liu, Y. Deep learning: A generic approach for extreme condition traffic forecasting. In Proceedings of the 2017 SIAM International Conference on Data Mining (SIAM 2017), Houston, TX, USA, 27–29 April 2017; pp. 777–785. [\[CrossRef\]](#)
- Zhang, J.; Zheng, Y.; Qi, D. Deep spatio-temporal residual networks for citywide crowd flows prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31. [\[CrossRef\]](#)
- Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; Ye, J.; Li, Z. Deep multi-view spatial-temporal network for taxi demand prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32. [\[CrossRef\]](#)
- Zhang, W.; Liu, H.; Liu, Y.; Zhou, J.; Xiong, H. Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 1186–1193. [\[CrossRef\]](#)

19. Wang, Y.; Wu, H.; Zhang, J.; Gao, Z.; Wang, J.; Philip, S.Y.; Long, M. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 2208–2225. [[CrossRef](#)]
20. He, S.; Luo, Q.; Du, R.; Zhao, L.; He, G.; Fu, H.; Li, H. STGC-GNNs: A GNN-based traffic prediction framework with a spatial–temporal Granger causality graph. *Phys. A Stat. Mech. Appl.* **2023**, *623*, 128913. [[CrossRef](#)]
21. Wang, Q.; Liu, W.; Wang, X.; Chen, X.; Chen, G.; Wu, Q. GMHANN: A Novel Traffic Flow Prediction Method for Transportation Management Based on Spatial-Temporal Graph Modeling. *IEEE Trans. Intell. Transp. Syst.* **2023**, *25*, 386–401. [[CrossRef](#)]
22. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926.
23. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
24. Park, C.; Lee, C.; Bahng, H.; Tae, Y.; Jin, S.; Kim, K.; Ko, S.; Choo, J. ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020; pp. 1215–1224. [[CrossRef](#)]
25. Wang, P.; Zhang, Y.; Hu, T.; Zhang, T. Urban traffic flow prediction: A dynamic temporal graph network considering missing values. *Int. J. Geogr. Inf. Sci.* **2023**, *37*, 885–912. [[CrossRef](#)]
26. Geng, X.; Li, Y.; Wang, L.; Zhang, L.; Yang, Q.; Ye, J.; Liu, Y. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 3656–3663. [[CrossRef](#)]
27. Lin, Z.; Feng, J.; Lu, Z.; Li, Y.; Jin, D. Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 1020–1027. [[CrossRef](#)]
28. Pan, Z.; Liang, Y.; Wang, W.; Yu, Y.; Zheng, Y.; Zhang, J. Urban traffic prediction from spatio-temporal data using deep meta learning. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1720–1730. [[CrossRef](#)]
29. Zhong, Z.; Li, C.T.; Pang, J. Hierarchical message-passing graph neural networks. *Data Min. Knowl. Discov.* **2023**, *37*, 381–408. [[CrossRef](#)]
30. Zhang, K.; Zhou, F.; Wu, L.; Xie, N.; He, Z. Semantic understanding and prompt engineering for large-scale traffic data imputation. *Inf. Fusion* **2024**, *102*, 102038. [[CrossRef](#)]
31. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. *arXiv* **2019**, arXiv:1906.00121.
32. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 922–929. [[CrossRef](#)]
33. Shang, C.; Chen, J.; Bi, J. Discrete graph structure learning for forecasting multiple time series. *arXiv* **2021**, arXiv:2101.06861. <https://doi.org/10.48550/arXiv.2101.06861>.
34. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional sequence to sequence learning. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; PMLR: London, UK, 2017; pp. 1243–1252. [[CrossRef](#)]
35. Chen, W.; Chen, L.; Xie, Y.; Cao, W.; Gao, Y.; Feng, X. Multi-range attentive bicomponent graph convolutional network for traffic forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 3529–3536. [[CrossRef](#)]
36. Nguyen, H.A.T.; Nguyen, H.D.; Do, T.H. An Application of Vector Autoregressive Model for Analyzing the Impact of Weather And Nearby Traffic Flow On The Traffic Volume. In Proceedings of the 2022 RIVF International Conference on Computing and Communication Technologies (RIVF), Ho Chi Minh City, Vietnam, 20–22 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 328–333. [[CrossRef](#)]
37. Liu, Y.; Zheng, H.; Feng, X.; Chen, Z. Short-term traffic flow prediction with Conv-LSTM. In Proceedings of the 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 11–13 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6. [[CrossRef](#)]
38. Bai, L.; Yao, L.; Li, C.; Wang, X.; Wang, C. Adaptive graph convolutional recurrent network for traffic forecasting. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17804–17815. [[CrossRef](#)]
39. Zhong, W.; Meidani, H.; Macfarlane, J. Attention-based Spatial-Temporal Graph Neural ODE for Traffic Prediction. *arXiv* **2023**, arXiv:2305.00985.
40. Liu, H.; Dong, Z.; Jiang, R.; Deng, J.; Deng, J.; Chen, Q.; Song, X. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, Birmingham, UK, 21–25 October 2023; pp. 4125–4129. [[CrossRef](#)]
41. Dai, B.A.; Ye, B.L. A novel hybrid time-varying graph neural network for traffic flow forecasting. *arXiv* **2024**, arXiv:2401.10155.
42. Zhu, Y.; Xu, W.; Zhang, J.; Du, Y.; Zhang, J.; Liu, Q.; Yang, C.; Wu, S. A survey on graph structure learning: Progress and opportunities. *arXiv* **2021**, arXiv:2103.03036.

43. Xue, G.; Zhong, M.; Li, J.; Chen, J.; Zhai, C.; Kong, R. Dynamic network embedding survey. *Neurocomputing* **2022**, *472*, 212–223. [[CrossRef](#)]
44. Cao, D.; Wang, Y.; Duan, J.; Zhang, C.; Zhu, X.; Huang, C.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17766–17778. [[CrossRef](#)]
45. Zhou, Z.; Zhou, S.; Mao, B.; Zhou, X.; Chen, J.; Tan, Q.; Zha, D.; Wang, C.; Feng, Y.; Chen, C. Opengsl: A comprehensive benchmark for graph structure learning. *arXiv* **2023**, arXiv:2306.10280.
46. Li, R.; Wang, S.; Zhu, F.; Huang, J. Adaptive graph convolutional neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32. [[CrossRef](#)]
47. Franceschi, L.; Niepert, M.; Pontil, M.; He, X. Learning discrete structures for graph neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; PMLR: London, UK, 2019, pp. 1972–1982. [[CrossRef](#)]
48. Zheng, C.; Zong, B.; Cheng, W.; Song, D.; Ni, J.; Yu, W.; Chen, H.; Wang, W. Robust graph representation learning via neural sparsification. In Proceedings of the International Conference on Machine Learning, Virtual, 12–18 July 2020; PMLR: London, UK, 2020; pp. 11458–11468.
49. Yang, L.; Kang, Z.; Cao, X.; Jin, D.; Yang, B.; Guo, Y. Topology Optimization based Graph Convolutional Network. In Proceedings of the International Joint Conference on Artificial Intelligence, Macau, China, 10–16 August 2019; pp. 4054–4061. [[CrossRef](#)]
50. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; Zhang, C. Connecting the dots: Multivariate time series forecasting with graph neural networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 753–763. [[CrossRef](#)]
51. Zügner, D.; Aubet, F.X.; Satorras, V.G.; Januschowski, T.; Günnemann, S.; Gasthaus, J. A study of joint graph inference and forecasting. *arXiv* **2021**, arXiv:2109.04979.
52. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
53. Bengio, S.; Vinyals, O.; Jaitly, N.; Shazeer, N. Scheduled sampling for sequence prediction with recurrent neural networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [[CrossRef](#)]
54. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; PMLR: London, UK, 2017; pp. 933–941.
55. Cai, C.; Wang, Y. A note on over-smoothing for graph neural networks. *arXiv* **2020**, arXiv:2006.13318.
56. Yu, H.; Li, T.; Yu, W.; Li, J.; Huang, Y.; Wang, L.; Liu, A. Regularized graph structure learning with semantic knowledge for multi-variates time-series forecasting. *arXiv* **2022**, arXiv:2210.06126.
57. Williams, B.M.; Hoel, L.A. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *J. Transp. Eng.* **2003**, *129*, 664–672. [[CrossRef](#)]
58. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*. [[CrossRef](#)]
59. Zheng, C.; Fan, X.; Wang, C.; Qi, J. Gman: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 1234–1241. [[CrossRef](#)]
60. Ye, J.; Sun, L.; Du, B.; Fu, Y.; Xiong, H. Coupled layer-wise graph convolution for transportation demand prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 4617–4625. [[CrossRef](#)]
61. Lee, H.; Jin, S.; Chu, H.; Lim, H.; Ko, S. Learning to remember patterns: Pattern matching memory networks for traffic forecasting. *arXiv* **2021**, arXiv:2110.10380.
62. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
63. Wang, G.; Ying, R.; Huang, J.; Leskovec, J. Multi-hop attention graph neural network. *arXiv* **2020**, arXiv:2009.14332.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.