ISSN: 1813-4890

Structure Semantic Aware Multi-Graph Network for Traffic Flow Forecasting

Linbiao Chen *

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China.

* chlinbiao@yeah.net

Abstract

Accurate traffic prediction plays a vital role in improving the safety, stability, and efficiency of intelligent transportation systems. However, existing methods often construct graphs that inherently entangle spatial and temporal information, failing to provide effective discriminative features. This limitation impairs the accurate representation of dynamic node interactions, thereby restricting the model's ability to capture traffic flow trends and periodic characteristics. To overcome this challenge, this paper introduces a novel Structure-Semantic Aware Multi-Graph Network (SSAMGN). First, we propose a graph learning module that jointly learns both the structural and semantic characteristics of the graph, adaptively generating sparse graphs to extract distinctive node features. Second, we develop a self-sampling approach that efficiently selects relevant historical sequences, coupled with a temporal-aware graph encoder that integrates temporal information into graph learning, thereby capturing unique temporal dependencies. Furthermore, we introduce an innovative temporal-aware attention mechanism, which leverages local contextual information to facilitate numerical sequence representation transformation, thereby enhancing the model's ability to learn traffic flow trends and periodic patterns. Extensive experiments on six publicly available datasets demonstrate that SSAMGN effectively alleviates the indistinguishability issue in graph representations and achieves state-of-the-art performance in traffic prediction tasks.

Keywords

Traffic flow forecasting; Graph neural networks; Temporal-aware graphs; Attention mechanism.

1. Introduction

In recent years, the rapid advancement of information technology and the continuous increase in motor vehicle ownership have significantly driven the development of intelligent transportation systems (ITS). Accurate traffic flow prediction plays a crucial role in reflecting real-time traffic conditions and assisting urban management departments in making informed scheduling decisions [1]. Moreover, various downstream tasks, such as travel time estimation, heavily rely on the accuracy of traffic flow prediction. As a result, traffic flow prediction has emerged as a key research focus in ITS.

Traffic flow data exhibit inherent time-series properties with complex temporal dependencies and dynamic spatial correlations. Temporal dependencies mainly arise from periodic and trend patterns, where traffic conditions at the same location show recurring similarities across days or within a day. Meanwhile, dynamic spatial correlations reflect mutual influences among different locations. As shown in Figure 1, relationships between nodes in a traffic network evolve over time, making static graph structures inadequate for capturing real-world

ISSN: 1813-4890

interactions. In practice, node connectivity should be dynamically updated based on real-time data. For example, strong spatial correlations emerge between areas near schools and business districts during morning rush hours, but weaken during off-peak periods. These observations highlight the intricate and nonlinear spatiotemporal characteristics of traffic flow data.

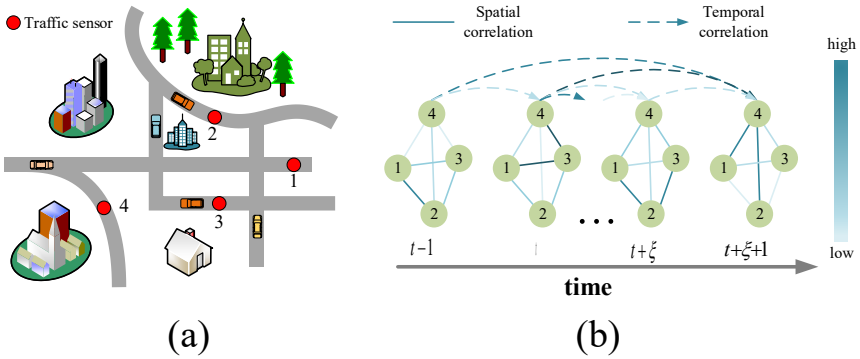

Fig.1 Complex spatiotemporal correlations. (a) A road network with loop detectors. (b) The corresponding spatiotemporal correlations of the loop detectors in (a). Although sensor 2 and sensor 4 are physically distant in the road network, at time step $t+\xi+1$, sensor 2 and sensor 4 are closely correlated, illustrating the dynamic long- and short-term spatial relationships in traffic flow. Furthermore, the traffic flow at sensor 4 during time step $t+\xi+1$ might be more closely related to a distant time step (e.g., $t-1$) than a nearby one (e.g., $t+\xi$), indicating the presence of complex nonlinear temporal relationships in traffic flow.

Traditional traffic flow prediction methods, such as the historical average (HA) model [2] and the autoregressive integrated moving average (ARIMA) model [3], rely heavily on idealized mathematical assumptions and are limited to modeling linear relationships. However, real-world traffic data are highly complex, dynamic, and nonlinear, making conventional approaches insufficient for modeling intricate spatiotemporal dependencies and dynamic spatial correlations in transportation networks. To overcome these limitations, grid-based methods have been proposed to extract spatiotemporal features by partitioning urban areas into regular grids [4][5]. Nevertheless, these methods impose linear structures on inherently nonlinear spatial relationships and ignore the true road network topology, limiting their ability to capture dynamic and nonlinear spatial features.

In recent years, with the rapid advancement of deep learning techniques, graph convolutional networks (GCNs) [6] have been increasingly employed for modeling non-Euclidean data. By aggregating information from neighboring nodes, GCNs enable deep spatial feature learning and extraction. Numerous studies have integrated GCNs with recurrent neural networks (RNNs) and convolutional neural networks (CNNs) to develop various traffic flow prediction models [7]. However, these methods often struggle to explicitly distinguish the interaction relationships of nodes across different time periods Figure 8. They typically classify graphs into either static or dynamic categories, assuming that dynamic graphs contain more valuable information than static ones [9][10], yet they overlook the fact that static and dynamic graphs represent structural and semantic information, respectively. Structural information encodes stable spatial correlations that form the foundation of graph topology, while semantic information captures time-dependent interactions such as intensified connections during peak hours. Overemphasizing either aspect limits the ability to learn rich and intrinsic spatiotemporal characteristics of traffic networks.

In recent years, with the rapid progress of deep learning, graph convolutional networks (GCNs) [6] have been increasingly employed for modeling non-Euclidean data. By aggregating information from neighboring nodes, GCNs enable deep spatial feature learning and extraction. Numerous studies have integrated GCNs with recurrent neural networks (RNNs) and convolutional neural networks (CNNs) to develop various traffic flow prediction models [7].

ISSN: 1813-4890

However, these methods often struggle to explicitly differentiate the interaction relationships of nodes across different time periods [8]. Existing approaches typically classify graphs as either static or dynamic, assuming that dynamic graphs provide more valuable information [9][10]. However, this assumption overlooks the complementary roles of static and dynamic graphs. Static graphs encode stable structural correlations among nodes, while dynamic graphs capture time-dependent semantic relationships, such as intensified interactions during peak hours. Overemphasizing either aspect limits a model's ability to learn rich and intrinsic spatiotemporal characteristics of traffic networks.

Additionally, most existing methods lack a thorough exploration of multi-scale temporal dependencies, limiting their capacity to represent complex temporal patterns in traffic flow. Many contemporary models [11][12] primarily rely on recent historical sequences, making them effective only for short-term dependencies [13] while failing to capture long-term temporal patterns. Moreover, several existing methods [14][15][16] employ fully connected or densely structured graphs, allowing unrestricted information propagation between nodes. While such architectures enhance information exchange, they also exacerbate the over-smoothing problem [17], where node embeddings become excessively similar, ultimately diminishing the model's ability to capture critical relationships.

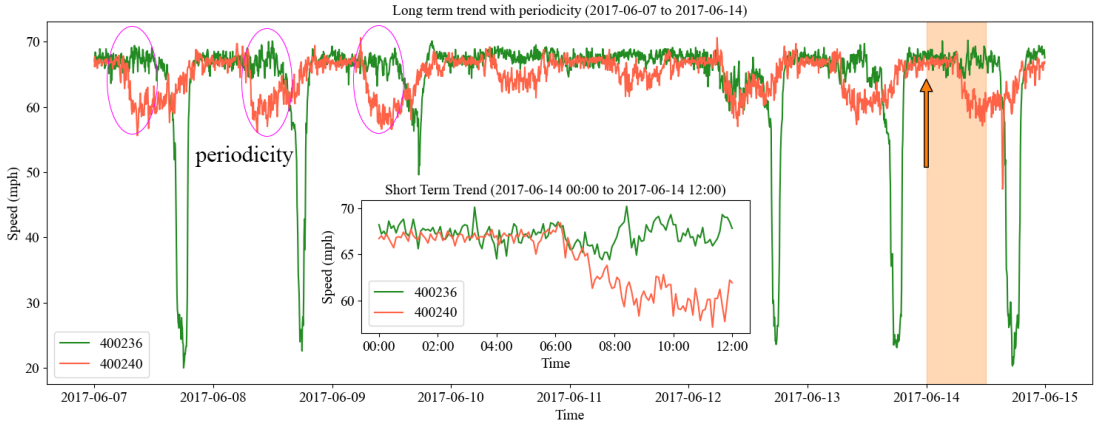

Fig.2 Comparative illustration of historical sequences from sensors 400236 and 400240 (PEMS-BAY dataset).

Most existing models rely on short-term inputs and thus fail to capture rich temporal characteristics in historical traffic data. As shown in Figure 2, predicting traffic flow after 06:00 using only six-hour historical data is challenging because short-term trends from different sensors are highly similar and lack sufficient discriminative information. In contrast, incorporating long-term historical sequences enables the model to exploit long-term trends and periodic patterns, as illustrated by the distinct recurring patterns in the lower panel of Figure 2. However, directly using all time steps of long historical sequences are computationally inefficient and introduces redundancy due to temporal continuity. Therefore, efficiently extracting key informative representations from long-term sequences is crucial for improving both prediction accuracy and efficiency.

To address the aforementioned challenges, this study proposes a Structure Semantic Aware Multi-Graph Network (SSAMGN). Specifically, a structure-semantic coupled graph learning (SSCGL) module is introduced to capture structural information via graph convolutional layers with shared structural parameters, while leveraging a graph attention mechanism to model node semantic relationships. Furthermore, a temporal-aware graph encoder (TAG Encoder) based on self-sampling is designed to effectively extract historical information that is highly correlated with future sequences. This encoder constructs historical, transformation, and future graphs to enable temporally-aware graph feature encoding. Additionally, to enhance the model's focus on critical dependency relationships, a sparse graph learning module is

developed to constrain the spatial receptive field and highlight node feature disparities. Moreover, a temporal trend-aware self-attention (TASAtt) module is incorporated to effectively capture the dynamic trends and periodic characteristics of the data.

The primary contributions of this study are summarized as follows:

1. A novel traffic flow prediction model, SSAMGN, is proposed to jointly learn both structural and semantic aspects of graphs, enabling the extraction of intrinsic graph characteristics and trend-aware traffic flow features.
2. A temporal-aware graph encoder based on a self-sampling strategy is introduced to sample relevant historical sequences and incorporate temporal information into graph learning, thereby capturing time-dependent features.
3. A sparse graph learning module is designed to enforce the identification of crucial and explicit connections while constraining the spatial receptive field to preserve the unique characteristics of nodes. Additionally, a trend-aware self-attention module has been designed to enable self-attention to perceive local contextual information.
4. Extensive experiments on six real-world datasets demonstrate that the proposed SSAMGN significantly outperforms state-of-the-art methods.

2. Related Works

2.1. Traffic flow forecasting

With the rapid advancement of transportation systems, accurate traffic flow forecasting has become increasingly crucial due to the intricate spatiotemporal dependencies inherent in traffic sequences. Traditional statistical methods, such as the autoregressive integrated moving average (ARIMA) model [18] and the vector autoregression (VAR) model [19], have been widely employed to analyze intrinsic patterns and stochastic perturbations in sequential data. However, these methods often struggle to capture complex nonlinear dependencies. In recent years, deep learning techniques have demonstrated remarkable success in sequence modeling. For instance, long short-term memory (LSTM) networks [20] utilize memory cells to retain historical information effectively. However, while LSTM-based models excel at capturing temporal dependencies, they primarily focus on individual time series and overlook crucial spatial interactions.

To address this limitation, an increasing number of studies have explored the potential of graph neural networks (GNNs) [21], incorporating techniques such as self-attention mechanisms [22] and graph convolutional networks (GCNs) [23] to capture spatial relationships among traffic nodes. Despite their promising performance, existing models mainly rely on short-term historical data to model local spatiotemporal dependencies and extract spatial information through graph structures, while largely ignoring the diversity of temporal patterns in traffic flow. The lack of multi-scale temporal modeling in graph learning leads to limited temporal awareness, which in turn causes graph indistinguishability and hinders the effective capture of complex temporal dynamics. Recent studies [24][25] have attempted to mitigate this issue by constructing static graphs using entire historical sequences, which remain unchanged throughout the forecasting process. However, such approaches fail to fully integrate diverse temporal dependencies into the graph-based learning framework. Additionally, models such as ST-WA [26] and STAR [27] have sought to extend the receptive field of historical sequences to mitigate short-term dependency limitations. However, these methods often incur high computational cost and struggle to capture long-range dynamic patterns. Efficiently integrating multi-scale temporal dependencies into graph-based learning therefore remains a key challenge in traffic flow forecasting.

ISSN: 1813-4890

2.2. Spatiotemporal graph neural networks

Spatiotemporal graph neural networks (STGNNs) effectively model spatial and temporal dependencies by integrating graph neural networks with sequence-to-sequence architectures, where graph construction plays a central role in characterizing latent spatiotemporal correlations. Early STGNN methods mainly relied on predefined graphs derived from real-world road networks. For example, DCRNN [8] and STGCN [9] leveraged graph convolution techniques to extract inter-node relationships based on static road networks. However, such predefined graphs may not accurately reflect the underlying spatial dependencies due to dynamic changes in traffic conditions. Consequently, data-driven approaches for learning adaptive graphs have gained significant attention. Graph WaveNet [14], for instance, integrates adaptive graph learning with graph convolution to dynamically capture hidden spatial dependencies, achieving state-of-the-art results. Building upon this foundation, models such as MTGNN [28], AGCRN [15], and STG-NCDE [29] have further enhanced adaptive graph modeling by incorporating node-specific and localized graph learning mechanisms. In addition, recent studies have underscored the limitations of static graph structures in representing dynamic traffic patterns, leading to the development of dynamic graph learning techniques. Methods such as GMAN [30] and ST-WA [26] employ self-attention mechanisms to dynamically update inter-node relationships, effectively overcoming the constraints of static graphs. Similarly, STDE-DGCN [31] constructs dynamic correlation graphs to capture fine-grained temporal dependencies between nodes.

However, a key challenge remains. Static graphs model topology, while dynamic graphs capture evolving semantics, and neglecting either limits intrinsic graph representation learning. Moreover, fully connected dynamic graphs often cause excessive information diffusion, degrading node-level features. Balancing structural consistency and dynamic adaptability therefore remains a core challenge in STGNN-based traffic forecasting.

3. Preliminary

3.1. Traffic Flow

Traffic sequences represent a specific type of multivariate time series that encapsulate traffic-related attributes, such as traffic flow, and consist of numerous interrelated variables or nodes. In this study, the traffic sequence is denoted as $X_{t+1:t+T} \in \mathbb{R}^{N\times T}$, where N represents the number of nodes and T denotes the number of time steps. The traffic sequence of node i is expressed as $X^{i}_{t+1:t+T} \in \mathbb{R}^{T}$.

3.2. Structural graph

Nodes in traffic networks exhibit structural connectivity based on their physical proximity within the road network. Consequently, the traffic network can be represented as a structural graph $G=\{V,E\}$, where $V=\{v_1,v_2,\cdots,v_N\}$ is the set of N nodes, and E represents the set of structural edges. The structural edge connecting nodes i and j is denoted as $e_{i,j}=\left(v_i,v_j,A_{i,j}\right)\in E$, where $A_{i,j}$ is the weight of the edge and corresponds to an element of the adjacency matrix A. Notably, due to the relative stability of the structural configuration of traffic systems, the structural graph remains unchanged over time.

3.3. Semantic graph

Nodes in a traffic system often exhibit semantic relationships even when they are not structurally connected. For instance, nodes located in different suburban residential areas may display similar traffic patterns during specific time periods. The semantic relationships between nodes at a given time step t can be modeled as a dynamic semantic graph $G_t=\{V,E_t\}$,

ISSN: 1813-4890

where E_t denotes the set of semantic edges at time step t. The semantic edge connecting nodes i and j at time step t is defined as $e_{i,j,t} = \left(v_i, v_j, A_{i,j,t}\right)$, where $A_{i,j,t} = \phi\left(x_t^i, x_t^j\right)$ quantifies the significance of the semantic relationship based on the traffic states of nodes i and j at timestamp t. The function ϕ is employed to generate semantic edges. Furthermore, the corresponding semantic adjacency matrix is denoted as A_t. Unlike the structural graph, the semantic graph is dynamic and evolves over time.

3.4. Problem of traffic flow forecasting

The objective of traffic flow forecasting is to predict the future sequence $Y_{t+1:t+T_f} \in \mathbb{R}^{N \times T_f}$ over the next T_f time steps based on the historical sequence $X_{t-T_h:t} \in \mathbb{R}^{N \times T_h}$ observed over the past T_h time steps. Additionally, an extended historical sequence of length S is sampled for node i, denoted as $\tilde{X}^i_{t'+1:t'+T_f}$ with $t' < t$, where $\tilde{X}^i_{t'+1:t'+T_f} \in \mathbb{R}^{S \times T_f}$. This sequence is considered relevant to the future sequence $Y^i_{t+1:t+T_f} \in \mathbb{R}^{T_f}$ of node i, and is incorporated as an additional input sequence.

4. Methodology

4.1. Overall architecture

Figure 3 presents the overall framework of SSAMGN. First, a self-sampling strategy extracts extended historical sequences relevant to the future sequence, which are aggregated into self-sampling embeddings, while the recent historical sequence is transformed into a recent historical embedding. Based on these embeddings, the TAG Encoder constructs three types of graphs, including historical, transition, and future graphs, to capture distinct temporal characteristics. The TAG Encoder consists of multiple SSCGL blocks that jointly learn structural and semantic information, with sparse graph propagation applied to extract diverse node features. The resulting representations are then processed by a trend-aware self-attention mechanism to incorporate local contextual information and enhance sequence representation. Finally, the time-aware graph embeddings are concatenated with the recent historical embeddings, self-sampling embeddings, and spatiotemporal features for traffic prediction.

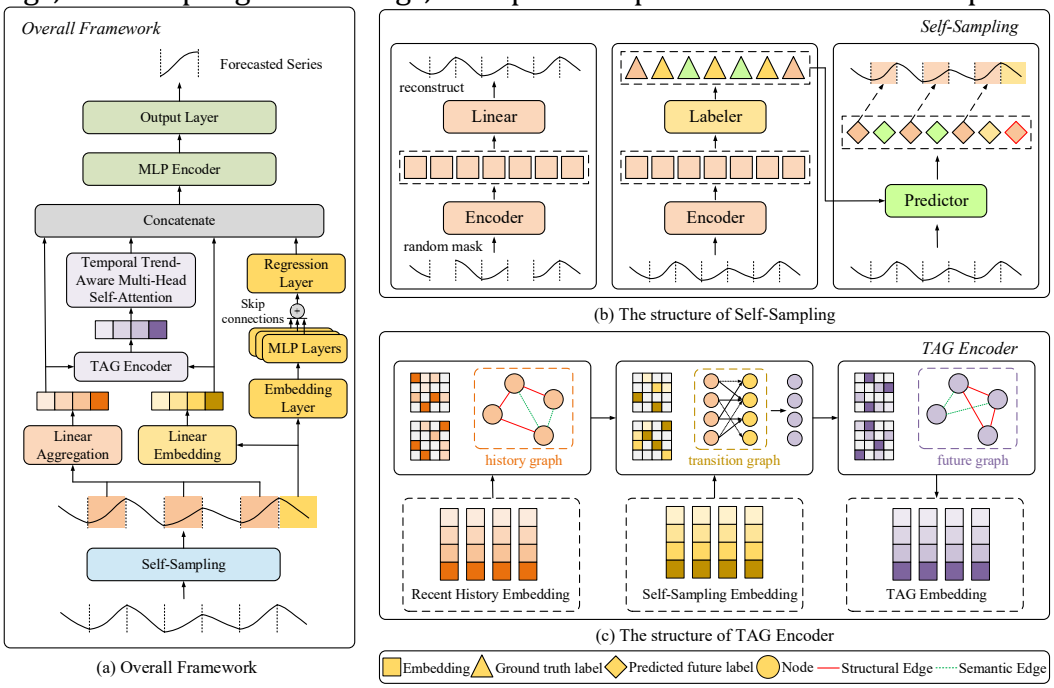

Fig.3 The overall framework of SSAMGN.

ISSN: 1813-4890

4.2. Self-sampling and Series Aggregation

To capture various temporal dependencies and incorporate time-related information into graph learning, this study proposes an adaptive sampling method, referred to as self-sampling, to select S historical fragments from past sequences that exhibit significant correlation with the target sequence $Y^i_{t+1:t+T_f} \in \mathbb{R}^{T_f}$. The parameter S is optimized through experimental tuning. However, directly computing the similarity between historical sequences and the target sequence incurs a computational complexity of $O(L^2)$, where L denotes the total length of the historical sequences in the dataset. To mitigate this issue, a self-supervised predictor is employed as a classifier to predict the labels of future sequences, enabling the selection of historically relevant sequences that share the same labels. This approach significantly reduces computational complexity to a more manageable $O(L)$.

The Self-Sampling mechanism primarily operates through the collaborative function of an encoder and a label generator, extracting spatiotemporal features and generating authentic labels for each sequence. The predictor is then trained using the most recent historical sequence $X^i_{t-T_h:t}$ to predict the label of the target sequence $Y^i_{t+1:t+T_f}$, thereby effectively sampling S historical sequences that share the same label as the target sequence. To ensure the selection of relevant sequences, capturing spatiotemporal semantics prior to sampling is crucial. To this end, an efficient and effective encoder is designed to extract spatiotemporal features. Initially, the raw sequence is randomly masked at a ratio r and linearly projected into a latent space:

$$H^i_{t+1:t+T_f} = FC_{embedding}\left(Y^{i'}_{t+1:t+T_f}\right) \tag{1}$$

where $Y^{i'}_{t+1:t+T_f} \in \mathbb{R}^{T_f}$ represents the masked sequence of node i, and $H^i_{t+1:t+T_f} \in \mathbb{R}^{D}$, with D being the hidden dimension.

Subsequently, spatial and temporal identity information E_i, T^{TiD}_t, and T^{TiW}_t are appended to the representation:

$$Z^{i(0)}_{t+1:t+T_f} = H^i_{t+1:t+T_f} \,\|\, E_i \,\|\, T^{TiD}_t \,\|\, T^{TiW}_t \tag{2}$$

where E_i, T^{TiD}_t, and T^{TiW}_t are learnable parameters corresponding to node identity, daily time, and weekly date, respectively. The symbol $\|$ denotes the concatenation operation between vectors. Following the inclusion of identity information, encoding is performed through L layers of residual multi-layer perceptron (MLP):

$$Z^{i(l+1)}_{t+1:t+T_f} = FC^l_2\left(ReLU\left(FC^l_1\left(Z^{i(l)}_{t+1:t+T_f}\right)\right)\right) + Z^{i(l)}_{t+1:t+T_f} \tag{3}$$

Finally, the embedding vector $Z^{i(L)}_{t+1:t+T_f} \in \mathbb{R}^{4D}$ is transformed via a regression layer to reconstruct the original sequence, ensuring that the embedding retains critical information:

$$\hat{Y}^i_{t+1:t+T_f} = FC_{regression}\left(Z^{i(L)}_{t+1:t+T_f}\right) \tag{4}$$

After obtaining spatiotemporal embeddings, relevant sequences are identified via unsupervised clustering. To avoid imbalance and the omission of rare temporal patterns, k-means with Euclidean distance is used for binary partitioning. The label generator is trained only on the training set and applied to both training and validation data to prevent information leakage.

Furthermore, during prediction, the future sequence $Y^i_{t+1:t+T_f}$ remains unseen, necessitating that its label be inferred solely from the most recent historical sequence $X^i_{t-T_h:t}$. Using the label C^i generated by the label generator as the ground truth, the predictor is trained to predict the label of the future sequence from the most recent historical sequence. The historical sequence $X^i_{t-T_h:t}$

ISSN: 1813-4890

is first processed by an embedding layer, generating the embedded vector $Z_{t-T_h}^{i(L)} \in \mathbb{R}^{4D}$. Given that sequences are clustered by nodes in the clustering stage, an MLP layer is applied to transform the output representation into the predicted label:

$$\hat{C}_{t+1:t+T_f}^{i} = argmax\left(FC_2^i\left(tanh\left(FC_1^i\left(Z_{t-T_h:t}^{i(L)}\right)\right)\right)\right) \tag{5}$$

where FC_1^i and FC_2^i denote the fully connected networks for node iii.

Following the label prediction for future sequences, the most recent S historical sequences that share the same predicted label with the future sequence are considered dependency sequences and are thus sampled as $\tilde{X}_{t'+1:t'+T_f}^{i}\left(t' < t\right) \in \mathbb{R}^{S \times T_f}$. The final sampled sequences are first independently linearly transformed into high-dimensional representations, then flattened and linearly aggregated to construct the self-sampling embedding:

$$H_{t'+1:t'+T_f}^{i,SS} = FC_{ss\_agg}\left(flatten\left(FC_{ss\_emb}\left(\tilde{X}_{t'+1:t'+T_f}^{i}\right)\right)\right) \tag{6}$$

where $H_{t'+1:t'+T_f}^{i,SS} \in \mathbb{R}^{D}$ represents the Self-Sampling embedding of node iii. Additionally, the most recent historical sequence of node i, $X_{t-T_h:t}^{i}$, is also linearly projected into the most recent historical embedding $X_{t-T_h:t}^{i,history} \in \mathbb{R}^{D}$.

4.3. Structure-Semantic-Coupled Graph Learning

The Structure-Semantic-Coupled Graph Learning (SSCGL) module is designed to enhance the structural-semantic coupling in graph learning, thereby improving the intrinsic feature extraction capability of graphs. The detailed structure is illustrated in Figure 4. The input graph embeddings are represented as the source embedding $H^{src} \in \mathbb{R}^{N \times D}$ and the target embedding $H^{tgt} \in \mathbb{R}^{N \times D}$, where N denotes the number of nodes. The SSCGL simulates information propagation from source to target embeddings by jointly leveraging structural and semantic graph information. To alleviate over-smoothing, it adopts sparse propagation to restrict receptive field diffusion, preserve node distinctiveness, and emphasize truly important connections.

First, information is propagated through a sparse graph convolution based on the graph structure:

$$H^{structure} = Softmax\left(sparsify\left(E_1 E_2^T, k\right)\right) H^{src} \tag{7}$$

where $H^{structure} \in \mathbb{R}^{N \times D}$, and $E_1, E_2^T \in \mathbb{R}^{N \times D}$ are learnable parameters representing the graph structure. The operation $(\square, k)$ sets all elements in each row to $-\infty$, except for the top-k elements, preventing node features from becoming indistinguishable during infinite propagation. Then, the propagation from source embedding to target embedding is performed based on the graph structure:

$$H^{tgt(1)} = LayerNorm\left(H^{tgt} + H^{structure}\right) \tag{8}$$

$$H^{tgt(2)} = LayerNorm\left(H^{tgt(1)} + FeedForward\left(H^{tgt(1)}\right)\right) \tag{9}$$

where $H^{tgt(1)}, H^{tgt(2)} \in \mathbb{R}^{N \times D}$. After structural propagation, a multi-head sparse attention mechanism is applied to capture semantic relationships:

$$Q_s = FC_{query}\left(H^{tgt(2)}\right), K_s = FC_{key}\left(H^{src}\right), V_s = FC_{value}\left(H^{src}\right) \tag{10}$$

$$H^{semantic} = Sparse\_Multihead\_Attention\left(Q_s, K_s, V_s\right) \tag{11}$$

where $Q_s, K_s, V_s, H^{semantic} \in \mathbb{R}^{N \times D}$. Subsequently, semantic information is propagated from the source embedding to the target embedding:

ISSN: 1813-4890

$$H^{tgt(3)} = LayerNorm\left(H^{tgt(2)} + H^{semantic}\right) \tag{12}$$

$$H^{tgt(4)} = LayerNorm\left(H^{tgt(3)} + FeedForward\left(H^{tgt(3)}\right)\right) \tag{13}$$

where $H^{tgt(3)}, H^{tgt(4)} \in \mathbb{R}^{N\times D}$, and $H^{tgt(4)}$ serves as the output embedding of the SSCGL module.

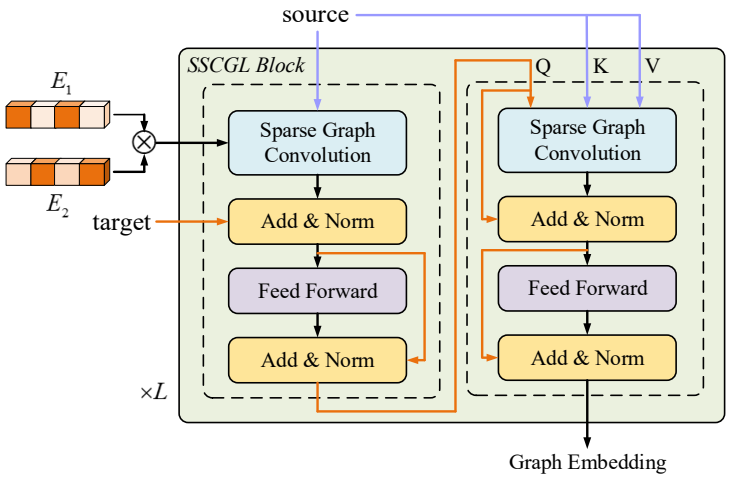

Fig.4 Structure of the SSCGL module.

4.4. Temporal-Aware Graphs Encoder

To overcome limited temporal receptive fields and overlooked temporal dependencies in graph learning, we propose the temporal-aware graphs encoder (TAG Encoder), as shown in Figure 5. By constructing historical, transition, and future graphs, the TAG Encoder simulates dynamic traffic interactions and injects temporal dependencies into graph learning. Integrated with semantic modeling through multiple SSCGL blocks, it effectively captures distinctive spatiotemporal graph features and alleviates graph indistinguishability.

Specifically, a linear transformation is first applied to the recent historical embeddings and the self-sampled embeddings of the nodes:

$$H^{history(0)}_{t-T_h:t} = FC_{src}\left(H^{history}_{t-T_h:t}\right) \tag{14}$$

$$H^{SS(0)}_{t'+1:t'+T_f} = FC_{tgt}\left(H^{SS}_{t'+1:t'+T_f}\right) \tag{15}$$

where $H^{history(0)}_{t-T_h}, H^{SS(0)}_{t'+1:t'+T_f} \in \mathbb{R}^{N\times D}$. These embeddings are further processed during learning and encoded using three types of graphs to extract comprehensive spatiotemporal features.

Historical Graph: The historical graph is inferred solely from recent historical embeddings to capture short-term spatiotemporal dependencies. The recent historical embeddings serve as both source and target embeddings and are transformed by the SSCGL block. Within the SSCGL module, the same graph parameters E^{src}_1 and E^{src}_2 are shared to construct the adjacency matrix in sparse graph convolution.

$$H^{history(l+1)}_{t-T_h:t} = SSCGL\left(H^{history(l)}_{t-T_h:t}, H^{history(l)}_{t-T_h:t}\right) \tag{16}$$

After applying B SSCGL blocks, the final graph embedding of the historical graph is denoted as $H^{history(B)}_{t-T_h:t}$.

Transition Graph: To model traffic flow transitions, a directed bipartite graph is constructed, where information propagates from recent historical embeddings to self-sampled embeddings to capture spatiotemporal dependencies during the transition process. Similar to the historical graph, SSCGL blocks are employed to capture spatial-temporal dependencies in the transitions, and the graph structure parameters E^{tgt}_1 and E^{src}_2 are shared across these blocks.

ISSN: 1813-4890

$$H_{t'+1:t'+T_f}^{SS(l+1)} = SSCGL\left(H_{t-T_h:t}^{history(B)}, H_{t'+1:t'+T_f}^{SS(l)}\right) \tag{17}$$

After B SSCGL blocks, the final graph embedding of the transition graph is denoted as $H_{t'+1:t'+T_f}^{SS(B)}$.

Future Graph: Node embeddings in the transition graph encode both short-term spatial dependencies and node-level temporal dependencies. To further propagate temporal information across nodes, a future graph is constructed to capture potential spatiotemporal dependencies under future traffic states, with its embeddings initialized from the transition graph embeddings.

$$H_{t+1:t+T_f}^{future(0)} = H_{t'+1:t'+T_f}^{SS(B)} \tag{18}$$

Subsequently, B SSCGL blocks with shared graph structure parameters E_1^{tgt} and E_2^{tgt} are applied:

$$H_{t+1:t+T_f}^{future(l+1)} = SSCGL\left(H_{t+1:t+T_f}^{future(l)}, H_{t+1:t+T_f}^{future(l)}\right) \tag{19}$$

Finally, the future graph embeddings are transformed into TAG embeddings:

$$H_{t+1:t+T_f}^{TAG} = FeedForward\left(H_{t+1:t+T_f}^{future(B)}\right) \tag{20}$$

where $H_{t+1:t+T_f}^{TAG} \in \mathbb{R}^{N\times D}$.

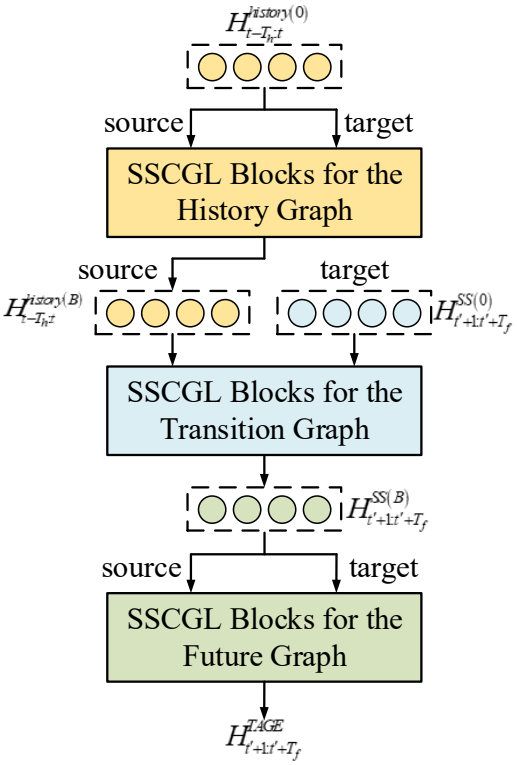

Fig.5 Temporal-Aware Graph Embedding Module.

4.5. Trend-Aware Self-Attention Mechanism

To model traffic flow trends and complexity, we design a trend-aware self-attention mechanism (TASAtt) that incorporates local contextual information. Self-attention computes queries, keys, and values from the same sequence, while multi-head self-attention extends this mechanism by attending to multiple representation subspaces in parallel. Its fundamental operation is defined as follows:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_{model}}}\right)V \tag{21}$$

ISSN: 1813-4890

where Q represents the query, K denotes the key, and V corresponds to the value.

Multi-head self-attention first projects the queries, keys, and values into multiple representation subspaces and applies the attention function in parallel. The resulting outputs are then concatenated and linearly projected to produce the final representation, which is formulated as follows:

$$Self-Attention(Q,K,V)=\oplus(head_1,\cdots,head_h)W^o \tag{22}$$

$$head_j=Attention(QW_j^Q,QW_j^K,QW_j^V) \tag{23}$$

where h represents the number of attention heads, W_j^Q , W_j^K , and W_j^V are the projection matrices for Q , K , and V , respectively, while W_j^O denotes the projection matrix for the final output. The multi-head self-attention mechanism provides a flexible approach to capturing the complex correlation dynamics in traffic data, thereby enabling accurate long-term forecasting.

However, standard multi-head self-attention is designed for discrete tokens and fails to capture local trend information in continuous sequences, making it less suitable for traffic signal modeling. To address this limitation, TASAtt introduces convolutional self-attention, where convolution operations explicitly encode local contextual features, enabling effective modeling of local variation trends in traffic data. Formally, TASAtt is defined as follows:

$$Trendhead_j=Attention(Q*\Phi_j^Q,K*\Phi_j^K,V*\Phi_j^V) \tag{24}$$

where Φ_j^Q and Φ_j^Q represent the convolutional kernel parameters.

The input $H_{t+1:t+T_f}^{TAG}$, and after operations on all nodes, the output is obtained as $H_{t+1:t+T_f}^{TASAtt}\in\mathbb{R}^{N\times D}$.

4.6. Forecasting Module

The proposed model employs a simple yet efficient MLP to generate predictions by integrating rich spatiotemporal information from multiple branches, including recent historical embeddings, self-sampling embeddings, TAG embeddings, spatiotemporal features, and trend-aware attention outputs. These multi-branch embeddings are first concatenated as follows:

$$Z_{t+1:t+T_f}^i=H_{t-T_h:t}^{i,history}\,\|\,H_{t'+1:t'+T_f}^{i,SS}\,\|\,H_{t+1:t+T_f}^{TASAtt}\,\|\,E_i\,\|\,T_t^{TiD}\,\|\,T_t^{TiW} \tag{25}$$

where $Z_{t+1:t+T_f}^i\in\mathbb{R}^{6D}$. The predicted sequence $\hat{Y}_{t+1:t+T_f}^i$ is then obtained using Equations (3) and (4).

5. Experiment

5.1. Datasets

Six publicly available traffic datasets were utilized in the experiments: METR-LA, PEMS-BAY, PEMS03, PEMS04, PEMS07, and PEMS08. These datasets were collected using loop detectors deployed on highways to obtain traffic data for corresponding road segments. Among them, METR-LA records traffic speed data from highways in Los Angeles, while PEMS-BAY contains traffic speed data from the San Francisco Bay Area. PEMS03, PEMS04, PEMS07, and PEMS08 consist of real-time traffic data collected by the Performance Measurement System (PeMS) of the California Department of Transportation at 30-second intervals. These datasets include information such as detection locations, detection dates, and data types. Detailed information on the experimental datasets is presented in Table 1.

Detailed information on the datasets. In the "Data Type" column, 'F' represents traffic flow, 'S'
Table 1 Denotes traffic speed, and 'O' indicates traffic occupancy.

Datasets	Nodes	Edges	Time Steps	Time interval	Data type	Time Range
METR-LA	207	1515	34272	5 min	S	03/2012-06/2012
PEMS-BAY	325	2369	52116	5 min	S	01/2017-05/2017
PeMS03	358	547	26208	5 min	F	09/2018-11/2018

PeMS04	307	340	16992	5 min	F, S, O	01/2018-02/2018
PeMS07	883	866	28224	5 min	F	05/2017-08/2017
PeMS08	170	259	17856	5 min	F, S, O	07/2016-08/2016

5.2. Experimental settings and hyperparameters

Traffic flow for the next 60 minutes ($T_f = 12$). The PEMS03, PEMS04, PEMS07, and PEMS08 datasets were divided into training, validation, and test sets in a ratio of 6:2:2, while the METR-LA and PEMS-BAY datasets were split in a 7:1:2 ratio. A self-sampling strategy was employed, with the sampling number S set to 7, determined through sensitivity analysis. When the historical sequence length was insufficient for sampling, the nearest historical sequence was used as a supplement to ensure the completeness of the input sequence. The hidden dimension D was set to 32, and the number of self-sampling encoding layers L was set to 4. To mitigate the randomness of clustering, each k-means algorithm was executed ten times, and the result with the smallest intra-cluster distance was selected. For the TAG Encoder, the number of SSCGL Blocks per graph was set to 2, and the sparsity parameter was set to 10. The training batch size was set to 32, and the Adam optimizer was used with a learning rate of 0.001. All experiments were conducted on a computing platform equipped with a 22 vCPU AMD EPYC 7T83 64-Core Processor and an RTX 4090 GPU.

5.3. Baselines

A comprehensive comparison was conducted between SSAMGN and various baseline models across different categories:

Traditional machine learning models: HA [18], ARIMA [18], VAR [19], and FC-LSTM [20].

Graph convolution-based models: GCRN [32], STGCN [9], STSGCN [33], DCRNN [8], Graph WaveNet [14], MTGNN [28], AGCRN [15], Z-GCNETs [34], STGODE [35], and STG-NCDE [29].

Attention-based models: ASTGCN(r) [36], GMAN [30], ASTGNN [13], DSTAGNN [24], ST-WA [26], and MSSTAT [25].

Non-graph-based models: STNorm [37] and STID [38].

5.4. Evaluation metrics

The performance of the model is evaluated using the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE), which are defined as follows:

$$\mathrm{MAE} = \frac{1}{K}\sum_{k=1}^{K}\left|X_{true(k)} - X_{pred(k)}\right| \tag{26}$$

$$\mathrm{MAPE} = \frac{1}{K}\sum_{k=1}^{K}\left|\frac{X_{true(k)} - X_{pred(k)}}{X_{true(k)}}\right| \tag{27}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(X_{true(k)} - X_{pred(k)}\right)^2} \tag{28}$$

where X_{true} represents the actual values, X_{pred} denotes the predicted values, and K is the number of samples. Lower values of MAE, RMSE, and MAPE indicate better predictive performance of the model.

5.5. Performance comparisons

In the traffic prediction experiments, the performance of the proposed SSAMGN model was evaluated on two types of benchmark datasets: traffic speed datasets (METR-LA and PEMS-BAY) and traffic flow datasets (PeMS03, PeMS04, PeMS07, and PeMS08). For METR-LA and PEMS-BAY, traffic speed predictions were conducted for time horizons of 15, 30, and 60 minutes.

ISSN: 1813-4890

Meanwhile, for PeMS03, PeMS04, PeMS07, and PeMS08, the experiments focused on predicting the average traffic flow over the next hour. The results of different models are summarized in Table 2Table 3 and Table 4, where the best performance is highlighted in bold, and the second-best results are underlined.

Overall, SSAMGN consistently outperformed all baseline models across all datasets, particularly in long-term forecasting tasks, demonstrating its strong ability to capture complex spatiotemporal dependencies and traffic trends. Traditional methods such as HA, VAR, and FC-LSTM exhibited inferior performance due to their limited capacity to model spatial dependencies. Although GCN-based models (e.g., STGCN and DCRNN) improve spatial modeling by incorporating predefined graphs, their reliance on fixed graph structures restricts their ability to reflect real-world spatial relationships. Adaptive graph-based models, including AGCRN and STG-NCDE, further enhance spatial representation but primarily focus on static graph structures, making them less effective in capturing dynamic semantic relationships among nodes. To address this, attention-based models such as GMAN and ST-WA introduce self-attention mechanisms to model dynamic dependencies; however, their inability to learn precise graph structures leads to suboptimal performance on datasets requiring fine-grained spatial modeling, such as PeMS03.

Despite these advances, existing methods generally overlook the inseparability of graph structures, as they tend to model either static topology or dynamic semantics independently. In contrast, SSAMGN explicitly incorporates time-dependent graph structure learning, enabling the joint modeling of traffic trends and complex spatiotemporal interactions, thereby achieving more accurate predictions. To further validate its effectiveness, visualization experiments were conducted on the PeMS04 and PeMS08 datasets, where SSAMGN was compared with STSGCN and ASTGCN(r) for 12-step-ahead forecasting, as shown in Figure 6. The results indicate that the prediction errors of STSGCN and ASTGCN(r) increase rapidly with longer horizons, reflecting error accumulation and limited long-term modeling capability. In comparison, SSAMGN exhibits more stable error growth across MAE, MAPE, and RMSE, highlighting its superior performance in long-term traffic forecasting..

Table 2 Traffic speed prediction results of different methods on METR-LA and PEMS-BAY

Datasets	Methods	15min			30min			60min		
		MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
METR-LA	HA	4.79	10.00	11.70	5.47	11.45	13.50	6.99	13.89	17.54
	ARIMA	3.99	8.21	9.60	5.15	10.45	12.70	6.90	13.23	17.40
	VAR	4.42	7.89	10.20	5.41	9.13	12.7	6.52	10.11	15.80
	FC-LSTM	3.44	6.30	9.60	3.77	7.23	10.90	4.37	8.69	14.00
	GCRN	3.03	5.75	8.26	3.54	6.92	10.11	4.32	8.48	13.05
	STGCN	2.88	5.74	7.62	3.47	7.24	9.57	4.59	9.40	12.70
	STSGCN	3.31	7.62	8.06	4.13	9.77	10.29	5.06	11.66	12.91
	DCRNN	2.77	5.38	7.30	3.15	6.45	8.80	3.60	7.59	10.50
	Graph WaveNet	**2.69**	**5.15**	6.90	3.07	6.22	8.37	3.53	7.37	10.01
	MTGNN	**2.69**	5.18	**6.88**	3.05	6.17	**8.19**	3.49	7.23	9.87
	AGCRN	2.85	5.53	7.63	3.20	6.52	9.00	3.59	7.45	10.47
	Z-GCNETs	3.23	7.48	7.87	3.93	9.40	9.75	4.83	11.57	12.04
	STGODE	3.47	6.76	8.76	4.36	8.47	11.14	5.50	10.33	14.32
	STG-NCDE	3.77	9.47	8.54	4.84	12.04	10.63	6.35	14.94	13.49
	ASTGCN(r)	4.86	9.27	9.21	5.43	10.61	10.13	6.51	12.52	11.64
	GMAN	2.80	5.55	7.41	3.02	6.49	8.73	3.44	7.35	10.07
	STNorm	2.81	5.57	7.40	3.18	6.59	8.47	3.57	7.51	10.24
	STID	2.82	5.53	7.75	3.19	6.57	9.39	3.55	7.55	10.95
	SSAMGN (ours)	2.70	5.28	7.18	**3.02**	**6.11**	8.25	**3.38**	**7.09**	**9.75**
PEMS-	HA	1.89	4.30	4.16	2.50	5.82	5.62	3.31	7.54	7.65

ISSN: 1813-4890

BAY	ARIMA	1.62	3.30	3.50	2.33	4.76	5.40	3.38	6.50	8.30
	VAR	1.74	3.16	3.60	2.32	4.25	5.00	2.93	5.44	6.50
	FC-LSTM	2.05	4.19	4.80	2.20	4.55	5.20	2.37	4.96	5.70
	GCRN	1.46	3.06	3.22	1.88	4.17	4.34	2.40	5.36	5.89
	STGCN	1.36	2.96	2.90	1.81	4.27	4.17	2.49	5.69	5.79
	STSGCN	1.44	3.01	3.04	1.83	4.18	4.17	2.26	5.21	5.40
	DCRNN	1.38	2.95	2.90	1.74	3.97	3.90	2.07	4.47	4.90
	Graph WaveNet	**1.30**	2.74	**2.73**	1.63	3.70	3.67	1.95	4.52	4.63
	MTGNN	1.32	2.79	2.77	1.65	3.74	3.69	1.94	4.49	4.53
	AGCRN	1.37	2.87	2.94	1.69	3.85	3.87	1.96	4.54	4.64
	Z-GCNETs	1.36	2.86	2.88	1.68	3.78	3.79	1.98	4.53	4.60
	STGODE	1.43	2.88	2.99	1.84	3.90	3.84	2.30	4.89	4.61
	STG-NCDE	1.38	2.93	2.91	1.71	3.84	3.91	2.03	4.58	4.82
	ASTGCN(r)	1.52	3.13	3.22	2.01	4.27	4.48	2.61	5.42	6.00
	GMAN	1.34	2.92	2.88	1.65	3.81	3.71	1.89	4.38	4.51
	STNorm	1.33	2.82	2.76	1.65	3.77	3.66	1.92	4.45	4.46
	STID	1.31	2.79	2.78	1.64	3.73	3.73	1.91	4.42	4.55
	SSAMGN (ours)	1.30	2.72	2.73	1.59	3.62	3.57	1.85	4.36	4.35

Table 3 Traffic flow prediction results of different methods on PeMS03 and PeMS07

Methods	PeMS03			PeMS07		
	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
GCRN	19.88	33.20	19.71	31.03	48.70	15.67
STGCN	17.55	30.42	17.34	25.33	39.34	11.21
STSGCN	17.48	29.21	16.78	24.26	39.03	10.21
DCRNN	17.99	30.31	18.34	25.22	38.61	11.82
Graph WaveNet	19.12	32.77	18.89	26.39	41.50	11.97
MTGNN	14.85	25.23	14.55	21.01	34.14	8.92
AGCRN	16.03	28.52	14.65	22.37	35.70	9.55
Z-GCNETs	16.64	28.15	16.39	21.77	35.17	9.25
STGODE	16.50	27.84	16.69	22.59	37.54	10.14
STG-NCDE	15.57	27.09	15.06	20.53	33.84	8.80
ASTGCN(r)	17.34	29.56	17.21	24.01	37.87	10.73
GMAN	16.87	27.92	18.23	20.43	33.30	8.69
ASTGNN	14.78	25.00	14.79	19.83	32.87	8.53
DSTAGNN	15.57	27.21	14.68	21.42	34.51	9.01
ST-WA	15.56	27.39	14.80	21.23	34.57	9.06
MSSTAT	15.35	25.39	15.66	19.78	32.85	8.73
STNorm	15.32	25.93	**14.37**	20.59	34.86	8.61
STID	15.33	27.40	16.40	19.54	32.85	8.25
SSAMGN (ours)	14.64	24.06	14.52	19.13	32.73	8.05

ISSN: 1813-4890

Table 4 Traffic flow prediction results of different methods on PeMS04 and PeMS08

Methods	PeMS04			PeMS08		
	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	24.54	38.61	17.24	19.19	29.81	13.10
FC-LSTM	26.77	40.65	18.23	23.09	35.17	14.99
GCRN	26.73	41.56	19.20	21.28	33.46	14.15
STGCN	21.16	34.89	13.83	17.50	27.09	11.29
STSGCN	21.19	33.65	13.90	17.13	26.80	10.96
DCRNN	21.22	33.44	14.17	16.82	26.36	10.92
Graph WaveNet	24.89	39.66	17.29	18.28	30.05	12.15
MTGNN	19.13	31.03	13.22	15.25	24.22	10.66
AGCRN	19.89	32.86	13.37	16.13	25.52	10.21
Z-GCNETs	19.50	31.61	12.78	15.76	25.11	10.01
STGODE	20.84	32.82	13.77	16.81	25.97	10.62
STG-NCDE	19.21	31.09	12.76	15.45	24.81	9.92
ASTGCN(r)	22.93	35.33	16.56	18.25	28.06	11.64
GMAN	18.83	30.93	13.21	14.81	24.19	9.69
ASTGNN	18.60	29.97	12.63	14.97	23.51	9.49
DSTAGNN	19.30	31.46	12.70	15.67	24.77	9.94
ST-WA	19.30	30.83	12.67	16.06	25.03	10.39
MSSTAT	18.57	30.37	12.23	**14.03**	<u>23.44</u>	9.34
STNorm	19.21	32.30	<u>13.05</u>	15.39	24.80	9.91
STID	<u>18.29</u>	<u>29.82</u>	**12.49**	14.20	23.49	<u>9.28</u>
SSAMGN (ours)	**18.26**	**29.77**	**12.49**	<u>14.08</u>	**23.36**	**9.24**

To provide a more intuitive evaluation of the proposed method, a 12-step-ahead prediction visualization was conducted on the PeMS04 and PeMS08 datasets, comparing SSAMGN with STSGCN and ASTGCN(r), as shown in Figure 6. As the forecasting horizon extends to 12 steps, STSGCN and ASTGCN(r) exhibit a pronounced increase in prediction errors, indicating their limited ability to capture long-term traffic flow trends. This suggests that most existing methods struggle to model long-term patterns and periodic characteristics of traffic data, resulting in error accumulation and degraded predictive accuracy. In contrast, SSAMGN consistently achieves lower MAE, MAPE, and RMSE, while maintaining a more stable error growth as the prediction horizon increases, demonstrating its superior long-term forecasting capability.

ISSN: 1813-4890

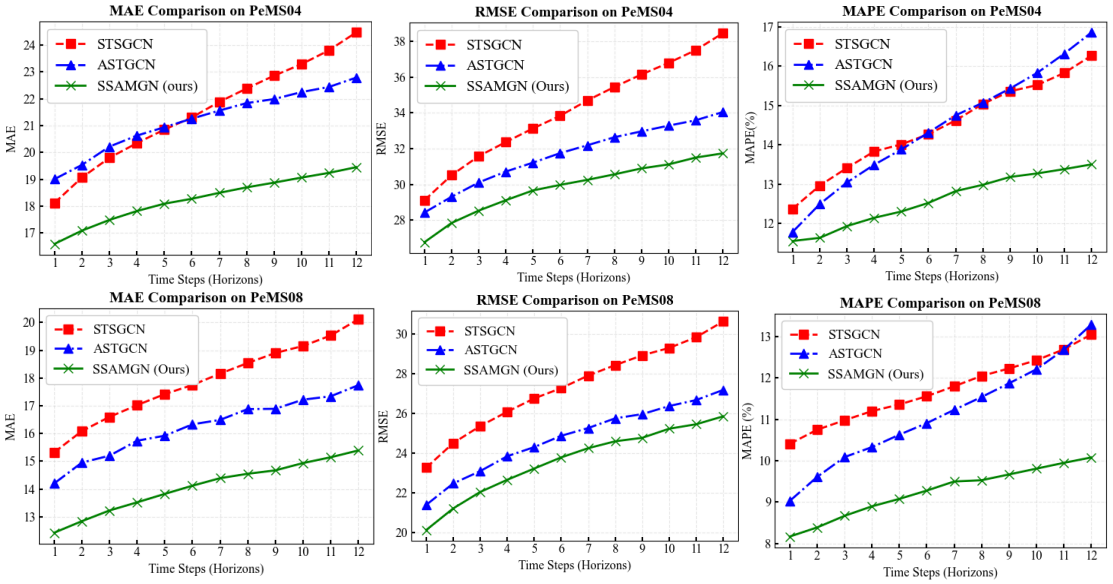

Fig.6 Comparison of prediction results for different methods across various horizons

5.6. Ablation experiments

The SSAMGN model consists of four core components with distinct functionalities: the SSCGL block jointly captures structural and semantic information from traffic graphs; the temporal-aware graph (TAG) module enables multi-scale temporal dependency modeling during graph learning; the Sparse Information Propagation mechanism improves interpretability and supports explicit graph structure learning; and the trend-aware self-attention (TASAtt) mechanism captures contextual sequence patterns to enhance long-term forecasting accuracy. The definitions of each variant are as follows:

w/o Structure Learning:The graph convolutional layers responsible for learning structural information in the SSCGL block were removed.

w/o Semantic Learning:The graph attention layers responsible for capturing semantic relationships in the SSCGL block were removed.

w/o Temporal-Aware Graphs: Only historical graphs were retained, while the transformation and future graphs in the TAG encoder were removed. Additionally, the historical graphs were constructed solely based on the most recent historical sequences.

w/o Sparsity:The learned graphs were fully connected and dense, where all nodes were completely connected.

w/o TASAtt:The trend-aware self-attention mechanism was removed from the model.

As shown in Table 5 and Figure 7, SSAMGN consistently outperforms all variants, demonstrating the effectiveness of each proposed component. Notably, removing structural learning causes more severe performance degradation than removing semantic learning, highlighting the fundamental role of structural information in traffic prediction. This suggests that structural graph connectivity provides the basis for semantic relationship modeling, as nodes with stronger structural connections tend to exhibit stronger semantic correlations. Moreover, the inferior performance of the w/o Temporal-Aware Graphs variant indicates that incorporating temporal dependencies substantially enhances spatial graph discrimination by expanding the temporal receptive field through the self-sampling strategy. The performance drop observed in the w/o Sparsity variant further confirms the importance of maintaining appropriate graph sparsity, since overly dense graphs dilute node-level attention and weaken critical relationships. Finally, the degradation in the w/o TASAtt variant verifies the essential role of trend-aware self-attention in capturing contextual information for long-term traffic forecasting.

ISSN: 1813-4890

Table 6. Traffic flow prediction results of different methods on PeMS04 and PeMS08

Architecture	PeMS03			PeMS04		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
w/o Structure Learning	14.91	25.35	15.13	18.51	30.15	13.24
w/o Semantic Learning	14.66	24.76	15.08	18.46	30.13	12.89
w/o Temporal-Aware Graphs	14.82	26.59	15.10	18.48	30.10	12.84
w/o Sparsity	14.72	24.81	15.05	18.53	30.16	13.13
w/o TASAtt	14.68	24.79	15.06	18.48	30.11	12.78
SSAMGN (ours)	14.64	24.06	14.52	18.26	29.77	12.49

Specifically, compared to the "w/o Structure Learning" model, SSAMGN shows improvements of 1.81%, 5.09%, and 4.04% in MAE, RMSE, and MAPE, respectively, on the PeMS03 dataset. On the PeMS04 dataset, it achieves improvements of 1.35%, 1.26%, and 5.67%, respectively. Similarly, when compared to the "w/o Semantic Learning" model, SSAMGN demonstrates improvements of 0.14%, 2.83%, and 3.72% on the PeMS03 dataset, and 1.08%, 1.19%, and 3.11% on the PeMS04 dataset. Although comparisons with other variants are omitted for brevity, the proposed modules significantly contribute to the overall performance improvement of SSAMGN.

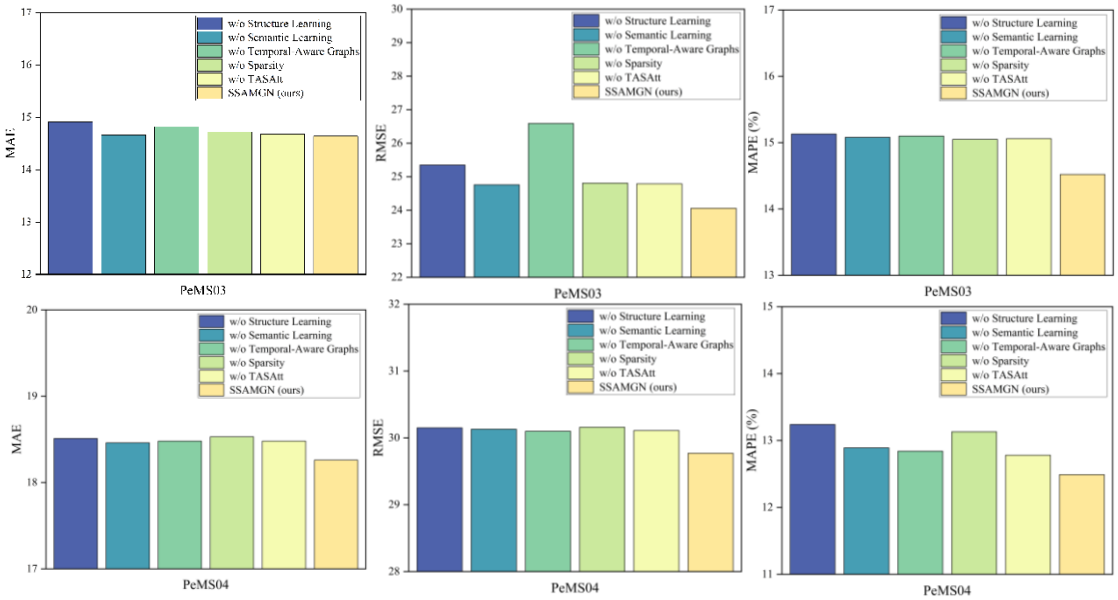

Fig.7 Component analysis of SSAMGN

5.7. Visualization

To further evaluate SSAMGN, a comparative visualization analysis between ground-truth traffic flow and predicted values was conducted across multiple datasets. As shown in Figure 8, both short-term forecasting with a prediction horizon of 3 and long-term forecasting with a horizon of 12 were analyzed. The results demonstrate that SSAMGN maintains strong consistency with actual traffic flows over different time spans and effectively captures peak-hour patterns and overall traffic trends, indicating a solid understanding of traffic dynamics and strong generalization ability.

On the PeMS03, PeMS04, and PeMS07 datasets, SSAMGN accurately captures sharp fluctuations during peak traffic periods, where many existing methods often fail. Although prediction errors increase slightly during highly volatile intervals, especially in long-term forecasting, the model still achieves reliable accuracy. Overall, these results confirm that SSAMGN can precisely model periodic traffic patterns while remaining robust to non-periodic variations and anomalous fluctuations across diverse traffic conditions.

ISSN: 1813-4890

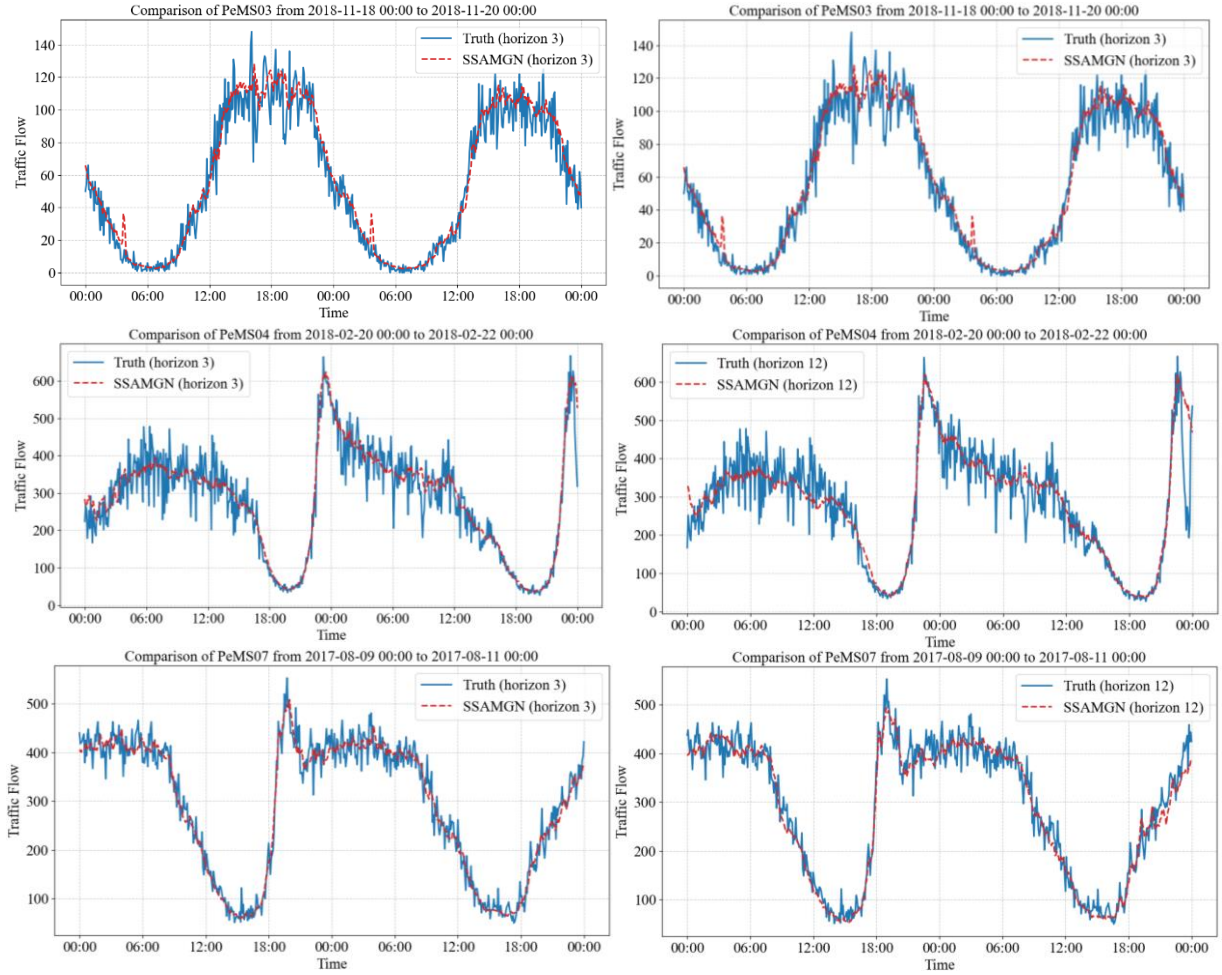

Fig.8 Visualization of actual and predicted values across different datasets

5.8. Spatial complexity study

Table 6 provides a summary of the total number of parameters for each model, while Figure 9 visualizes the relationship between parameter size and RMSE on the PeMS08 dataset. Notably, the SSAMGN model has a parameter size of only 1,120,174 bytes, which is significantly smaller than traditional models such as ASTGCN(r) and STGCN and even slightly more compact than DCRNN. As depicted in Figure 9, SSAMGN maintains a competitive balance between a relatively small parameter size and high predictive accuracy, underscoring its advantage in spatial complexity. These findings validate that the proposed model can achieve superior predictive performance while maintaining computational efficiency, making it well-suited for deployment in resource-constrained environments.

Table 6. Traffic flow prediction results of different methods on PeMS04 and PeMS08

Methods	ASTGCN(r)	DCRNN	STGCN	SSAMGN
Total Parameters (Byte)	2,251,496	1,187,844	2,171,140	**1,120,174**

ISSN: 1813-4890

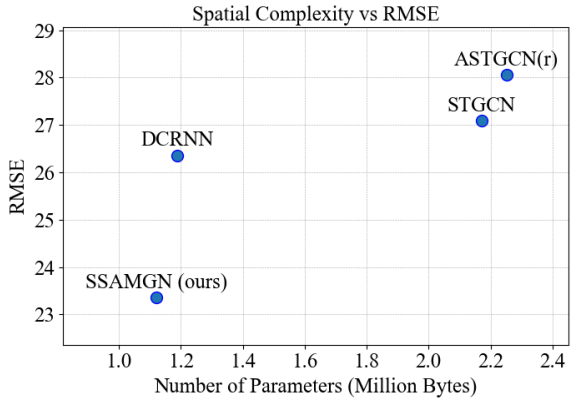

Fig.9 Comparison of the total number of model parameters and RMSE

6. Conclusion

This study presents SSAMGN, a novel traffic flow prediction model designed to address graph indistinguishability in traffic forecasting. By integrating graph structural information with semantic features through joint learning, the model enhances the representation of network characteristics. SSAMGN employs an adaptive sampling strategy and a temporal-aware graph encoder to capture multi-scale temporal dependencies while leveraging a constrained spatial receptive field to mitigate over-smoothing, ensuring more refined node feature extraction. Additionally, the trend-aware multi-head attention mechanism improves the capture of local contextual information, enhancing the model's ability to identify complex temporal trends in traffic data. Experiments on six real-world datasets demonstrate that SSAMGN consistently outperforms state-of-the-art baselines, achieving superior prediction accuracy.

Future work will focus on optimizing spatiotemporal graph construction and incorporating external factors such as weather, traffic incidents, and holidays to further improve prediction robustness.

Acknowledgements

This work is supported by the Key R&D Program of Gansu Province (23YFGA0063); the National Natural Science Foundation of China (62363022,61663021); the Natural Science Foundation Key Projects of Gansu Province (22JR5RA226); the Natural Science Foundation of Gansu Province (22JR5RA226, 23JRRA886).

References

[1] Jin, D., Shi, J., Wang, R., Li, Y., Huang, Y., & Yang, Y.-B. (2023). Trafformer: Unify time and space in traffic prediction. Proceedings of the AAAI Conference on Artificial Intelligence, 37(7), 8114-8122. https://doi.org/10.1609/aaai.v37i7.25980.

[2] Wei, G. (2004). A summary of traffic flow forecasting methods. Journal of Highway and Transportation Research and Development, 21(3), 82-85. Retrieved from https://api.semanticscholar.org/CorpusID:63829738.

[3] Comert, G., & Bezuglov, A. (2013). An online change-point-based model for traffic parameter prediction. IEEE Transactions on Intelligent Transportation Systems, 14(3), 1360-1369. https://doi.org/10.1109/TITS.2013.2260540.

[4] Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., & Zhang, C. (2018). Disan: Directional self-attention network for RNN/CNN-free language understanding. Proceedings of the AAAI Conference on Artificial Intelligence, 32. https://doi.org/10.48550/arXiv.1709.04696.

[5] Zhang, J., Zheng, Y., Sun, J., & Qi, D. (2020). Flow prediction in spatio-temporal networks

based on multitask deep learning. IEEE Transactions on Knowledge and Data Engineering, 32(3), 468-478. https://doi.org/10.1109/TKDE.2019.2891537.

[6] Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., & Li, H. (2020). T-GCN: A temporal graph convolutional network for traffic forecasting. IEEE Transactions on Intelligent Transportation Systems, 21(9), 3848-3858. https:// doi.org/ 10.1109/ TITS. 2019.2935152.

[7] Zhu, J., Wang, Q., Tao, C., Deng, H., Zhao, L., & Li, H. (2021). AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting. IEEE Access, 9, 35973–35983. https://doi.org/10.1109/ACCESS.2021.3062114.

[8] Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1707.01926.

[9] Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. Proceedings of the 27th International Joint Conference on Artificial Intelligence, 3634-3640. https:// doi.org/ 10.24963/ ijcai. 2018/505.

[10] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. International Conference on Learning Representations. https:// doi. org/ 10.48550/arXiv.1710.10903.

[11] Li, H., Yang, S., Luo, Y., Li, J., Song, Y., & Zhou, T. (2022). Spatial dynamic graph convolutional network for traffic flow forecasting. Applied Intelligence. https://doi.org/10.1007/ s10489-022-04271-z.

[12] Geng, X., Li, Y., Wang, L., Zhang, L., Yang, Q., Ye, J., & Liu, Y. (2019). Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 3656-3663. https://doi.org/10.1609/aaai.v33i01. 33013656.

[13] Guo, S., Lin, Y., Wan, H., Li, X., & Cong, G. (2022). Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. IEEE Transactions on Knowledge and Data Engineering, 34(11), 5415-5428. https://doi.org/10.1109/TKDE.2021.3056502.

[14] Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph WaveNet for deep spatial-temporal graph modeling. Proceedings of the 28th International Joint Conference on Artificial Intelligence, 1907-1913. https://doi.org/10.5555/3367243.3367303.

[15] Bai, L., Yao, L., Li, C., Wang, X., & Wang, C. (2020). Adaptive graph convolutional recurrent network for traffic forecasting. Advances in Neural Information Processing Systems, 33, 17804-17815. https://doi.org/10.48550/arXiv.2007.02842.

[16] Huang, R., Huang, C., Liu, Y., Dai, G., & Kong, W. (2020). LSGCN: Long short-term traffic prediction with graph convolutional networks. Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2355-2361. https://doi. org/10. 5555/ 3491440.3491766.

[17] Pu, B., Liu, J., Kang, Y., Chen, J., & Yu, P. S. (2024). MVSTT: A multiview spatial-temporal transformer network for traffic-flow forecasting. IEEE Transactions on Cybernetics, 54(3), 1582-1595. https://doi.org/10.1109/TCYB.2022.3223918.

[18] Wan Ahmad, W. K. A., & Ahmad, S. (2013). ARIMA model and exponential smoothing method: A comparison. AIP Conference Proceedings, 1522, 1312-1321. https:// doi.org/ 10.1063/ 1.4801282.

[19] Zivot, E., & Wang, J. (2003). Vector autoregressive models for multivariate time series. Springer, New York, NY. https://doi.org/10.1007/978-0-387-32348-0_11.

[20] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[21] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The Graph

Neural Network model. IEEE Transactions on Neural Networks, 20(1), 61-80. https://doi.org/ 10.1109/TNN.2008.2005605.

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30. https://doi.org/10.48550/arXiv.1706.03762.

[23] Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks: A comprehensive review. Computational Social Networks, 6(1), 1-23. https://doi.org/ 10.1186/ s40649-019-0069-y.

[24] Lan, S., Ma, Y., Huang, W., Yang, H., & Li, P. (2022). DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. International Conference on Machine Learning, 11906-11917. Retrieved from https://api. semanticscholar.org/ CorpusID:250.

[25] Tian, R., Wang, C., Hu, J., & Ma, Z. (2023). Multi-scale spatial-temporal aware transformer for traffic prediction. Information Sciences, 648, 119557. https://doi.org/ 10.1016/ j.ins.2023.119557.

[26] Cirstea, R.-G., Yang, B., Guo, C., Kieu, T., & Pan, S. (2022). Towards spatio-temporal aware traffic time series forecasting. 2022 IEEE 38th International Conference on Data Engineering (ICDE), 2900-2913. https://doi.org/10.1109/ICDE53745.2022.00262.

[27] Liang, W., Li, Y., Xie, K., Zhang, D., Li, K.-C., Souri, A., & Li, K. (2022). Spatial-temporal aware inductive graph neural network for C-ITS data recovery. IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/TITS.2022.3156266.

[28] Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., & Zhang, C. (2020). Connecting the dots: Multivariate time series forecasting with graph neural networks. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 753-763. https://doi.org/10.1145/3394486.3403118.

[29] Choi, J., Choi, H., Hwang, J., & Park, N. (2021). Graph neural controlled differential equations for traffic forecasting. arXiv preprint arXiv:2112.03558. https://doi.org/ 10.48550/ arXiv.2112. 03558.

[30] Zheng, C., Fan, X., Wang, C., & Qi, J. (2020). GMAN: A graph multi-attention network for traffic prediction. Proceedings of the AAAI Conference on Artificial Intelligence, 34(1), 1234-1241. https://doi.org/10.48550/arXiv.1911.08415.

[31] Zhang, W., Zhu, K., Zhang, S., Chen, Q., & Xu, J. (2022). Dynamic graph convolutional networks based on spatiotemporal data embedding for traffic flow forecasting. Knowledge-Based Systems, 250, 109028. https://doi.org/10.1016/j.knosys.2022.109028.

[32] Seo, Y., Defferrard, M., Vandergheynst, P., & Bresson, X. (2018). Structured sequence modeling with graph convolutional recurrent networks. International Conference on Neural Information Processing, 362-373. https://doi.org/10.1007/978-3-030-04167-0_33.

[33] Song, C., Lin, Y., Guo, S., & Wan, H. (2020). Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 34(1), 914-921. https://doi.org/10.1609/aaai.v34i01. 5438.

[34] Chen, Y., Segovia, I., & Gel, Y. (2021). Z-GCNETs: Time zigzags at graph convolutional networks for time series forecasting. International Conference on Machine Learning, 1684-1694. https://doi.org/10.48550/arXiv.2105.04100.

[35] Fang, Z., Long, Q., Song, G., & Xie, K. (2021). Spatial-temporal graph ODE networks for traffic flow forecasting. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 364-373. https://doi.org/10.1016/j.eswa.2023.122381.

[36] Guo, S., Lin, Y., Feng, N., Song, C., & Wan, H. (2019). Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. Proceedings of the AAAI Conference on

Artificial Intelligence, 33(1), 922-929. https://doi.org/10.1609/aaai.v33i01.3301922.
[37]Deng, J., Chen, X., Jiang, R., Song, X., & Tsang, I. W. (2021). ST-Norm: Spatial and temporal normalization for multi-variate time series forecasting. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 269-278. Retrieved from http://hdl.handle. net/10453/153747.
[38]Shao, Z., Zhang, Z., Wang, F., Wei, W., & Xu, Y. (2022). Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 4454-4458. https:// doi.org/10.1145/3511808.3557702.