A Multilayer Spatiotemporal Correlation-Aware Graph Attention Network for Traffic Flow Prediction

Junjie Liu, Yu Wang, Jiaxian Zhu, Weihua Bai, Huibing Zhang, Liyun Zuo, Teng Zhou, *Member, IEEE*, and Keqin Li, *Fellow, IEEE*

***Abstract*—Traffic flow prediction is fundamental to traffic information services, control, and guidance. The challenge to accurately model the traffic flow is to comprehensively capture dynamic, global spatial, and local temporal correlations. To address challenges in spatiotemporal dependencies, global similarity, and local dynamics, we propose a multilayer spatiotemporal correlation-aware graph attention network (MSTC-GAT) for traffic flow prediction. Our model contains a multilayer spatial structure-aware module [spatial graph attention network (S-GAT)] using a spatial GAT with hierarchical attention masks and a path-based node correlation matrix to effectively capture local and global spatial dependencies. The temporal structure-aware module [temporal graph attention networks (T-GATs)] constructs a short-term similarity matrix of nodes for the temporal GAT to capture local dynamic temporal dependencies. Finally, a spatiotemporal Transformer (ST-Transformer) fuses weighted spatiotemporal node embeddings to capture global dynamic dependencies for accurate prediction. We conduct extensive experiments on four public benchmark datasets compared with 10 state-of-the-art models. The experimental results demonstrate that the MSTC-GAT outperforms all comparisons for short- and long-term predictions.**

***Index Terms*—Contrastive learning, graph deep learning, heterogeneous graph, knowledge tracing, meta-path, personalized learning systems.**

Received 9 October 2024; revised 12 May 2025 and 10 August 2025; accepted 6 November 2025. Date of publication 13 November 2025; date of current version 4 May 2026. This work was supported in part by the National Natural Science Foundation under Grant 62462021, Grant 62267003, and Grant 62177012; in part by the Philosophy and Social Sciences Planning Project of Zhejiang Province under Grant 25JCXK006YB; in part by Hainan Provincial Natural Science Foundation of China under Grant 625RC716; in part by Guangxi Key Laboratory of Trusted Software under Grant KX202319; in part by the Special Fund for Guangdong Province University Key Field under Grant 2023ZDZX3041 and Grant 2023ZDZX3013; in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2025A1515010197 and Grant 2024A1515010144; in part by Zhaoqing University Innovation Research Team Grant Project; and in part by the Innovation Project of Guangdong Province University under Grant 2024KQNCX023. *(Corresponding authors: Yu Wang; Teng Zhou.)*

Junjie Liu, Jiaxian Zhu, and Weihua Bai are with the School of Computer Science and Software, Zhaoqing University, Zhaoqing, Guangdong 526061, China (e-mail: liujunjie@zqu.edu.cn; zhujiaxian@zqu.edu.cn; baiweihua@zqu.edu.cn).

Yu Wang and Liyun Zuo are with the School of Computer Science, Guangdong University of Petrochemical Technology, Maoming 525000, China (e-mail: wangy@gdupt.edu.cn; zuoliyun@gdupt.edu.cn).

Huibing Zhang is with Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: zhanghuibing@guet.edu.cn).

Teng Zhou is with the School of Cyberspace Security (School of Cryptology), Hainan University, Haikou 570228, China, and also with the Yangtze Delta Region Institute, University of Electronic Science and Technology of China, Quzhou 324003, China (e-mail: teng.zhou@hainanu.edu.cn).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TNNLS.2025.3630903

NOMENCLATURE

Notations	Definitions
$\mathbf{X}^{(t-T+1:t)}$	Historical traffic flow.
$G_s = (V_s, \text{Edge}_s)$	Undirected graph.
$\hat{Y}^{(t+1:t+T_p)}$	Final prediction.
$S_T(G)$	Structural entropy of the encoding tree T.
$S^{\text{cor}}_{T^*}(\alpha;\beta)$	Structural entropy between nodes α and β.
L_{T^*}	Maximum height of the encoding tree.
$\mathbf{M}^{(H)}_{T^*}$	Path-based hierarchical relative structural entropy matrix.
$\mathbf{A}_s$	Set of adjacency matrix.
M_{sim}	Node temporal similarity matrix.

I. INTRODUCTION

INTELLIGENT transport systems have been recognized as a crucial means to address urban traffic congestion and enhance travel safety [1]. As a core component of intelligent transportation systems (ITSs), traffic flow prediction serves as the foundation for traffic information services, traffic control, and guidance. It utilizes dynamic traffic data to forecast future traffic flow within the road network. Traditional traffic flow prediction uses sensors to collect flow data at critical road nodes. Real-time and accurate prediction systems are essential for alleviating traffic pressure, planning vehicle travel, and ensuring efficient and safe operation of transportation systems. By monitoring traffic flow on specific road segments and automatically predicting short-term flow changes, such systems can optimize traffic signal timing, alleviate congestion, improve traffic efficiency, and reduce accidents, thereby lowering travel costs and enhancing the quality of life.

Conventional traffic flow prediction methods have primarily focused on temporal dynamic modeling methodologies [2] and machine learning techniques [3], [4], which have demonstrated efficacy in the temporal dimension. However, these approaches exhibit limitations in addressing the spatiotemporal dependencies inherent in complex road networks due to their insufficient consideration of the intricate structures influenced by spatial dependencies. Although grid-based deep learning methods [5], [6] can capture certain spatiotemporal correlations [7], [8], they may lead to logical inconsistencies when representing the topological structure of actual road networks. While traditional methods possess advantages in handling temporal dynamics, they face challenges in effectively managing complex spatiotemporal dependencies.

In recent years, traffic flow prediction methods incorporating graph neural networks (GNNs), which combine graph theory and deep learning, have emerged as a focal point of research. These approaches represent traffic networks through graph structures, accurately reflecting the topological associations between roads and the dynamic changes in traffic flow. Complex spatial relationships are represented through adjacency matrices, and when combined with temporal information, GNNs can capture the dynamic spatial and temporal dependencies of traffic flow, such as GL-STGTN [8], STGCN [9], DCRNN [10], TWDGCN [11], and DDSTGCN [12]. The dynamic spatiotemporal gated Transformer GNN is particularly notable for its ability to simultaneously capture both local and global spatiotemporal dependencies [13]. The STGCN integrates graph latent information learning modules with traffic prediction network modules, dynamically updating adjacency matrices to adapt to changes in node attributes (e.g., velocity and flow). However, existing spatiotemporal traffic flow prediction models still face the following challenges: 1) predefined graph structures limit their ability to adapt to real-time network topology and flow fluctuations; 2) local dynamic temporal dependencies induced by external factors such as weather conditions, accidents, and road maintenance are not effectively integrated; and 3) the capture of short-term and localized features lacks precision, affecting the accuracy of short-term decision-making.

In the study of temporal graphs, the locality, globality, and correlation of space, along with the dynamics and dependencies of time and space, are crucial for modeling spatiotemporal data analysis [14], [15]. Integrating these features enhances models' predictive performance in complex environments and provides a theoretical foundation for the development of more refined spatiotemporal prediction models [16], [17]. However, capturing intricate dynamic patterns presents significant challenges, primarily due to the computational complexity and the difficulty in interpreting the proposed models [18]. To address these challenges, Wang et al. [19] proposed the global and local embedding network (GLEN), which not only dynamically generates node embeddings but also extracts higher-order semantic relationships through a cross-perspective fusion module. This approach improves interpretability by providing clearer insights into the relationships between features. Additionally, the dynamic diffusion-variational GNN (DVGNN) constructs dynamic graphs using an unsupervised generative model, focusing on causal relationships between neighboring nodes and combining dynamic graph convolution networks with a temporal attention mechanism to effectively capture temporal information and reconstruct causal graphs [20]. This model reduces computational costs by streamlining the graph construction process. Furthermore, the inadequacies of spatial graph structures and the complexities of spatiotemporal data must be resolved [21], [22]. Zhang et al. [23] introduced the spatial–temporal fusion adaptive gated graph convolution networks (STFAG-GCNs), which model missing information through spatial–temporal fusion graph construction and utilize a gated adaptive fusion graph convolution mechanism to dynamically model spatial–temporal correlations. Lastly, Wu et al. [24] proposed TraverseNet, which employs a message-passing mechanism to treat space and time as an integrated whole, mining spatiotemporal patterns and enhancing learning outcomes, thereby comprehensively evaluating the temporal dependencies of nodes on their neighbors.

To address the challenges of spatiotemporal dependencies, global similarities, and local dynamics in traffic flow prediction, we propose a novel traffic flow prediction model with multilayer spatiotemporal structure-dependent awareness, termed the multilayer spatiotemporal correlation-aware graph attention network (MSTC-GAT). Our model can capture complex spatial and temporal dependencies through two key components, that is, multilayer spatial structure-aware modules [spatial graph attention networks (S-GATs)] and temporal structure-aware modules [temporal graph attention networks (T-GATs)].

The S-GAT module constructs a graph structure correlation matrix, which is generated using an encoding tree by minimizing structural entropy on the adjacency matrix. This matrix incorporates hierarchy-based attention masks and path-based node correlation matrices, which are used as the input for the spatial graph attention neural network. Conversely, the T-GAT module computes node temporal similarities based on time slices to construct a short-term node similarity matrix, which is used as the input for the temporal graph attention neural network.

Finally, the spatiotemporal Transformer (ST-Transformer) captures global dynamic dependencies during the spatiotemporal feature fusion stage to produce the prediction results. The MSTC-GAT model addresses the intricate spatiotemporal relationships inherent in traffic flow prediction tasks.

The main contributions are summarized as follows.

1) *Multilayer Spatiotemporal Awareness:* Our model incorporates multilayer spatial and temporal awareness modules (S-GAT/T-GAT) to capture complex spatial and temporal dependencies to enhance traffic flow modeling.
2) *Novel Graph Structure Representation:* We propose an innovative graph structure representation by utilizing an encoding tree based on minimized structural entropy to construct hierarchical and relative path-based spatial correlation matrices. This approach significantly enhances the model's comprehension of complex graph structures and its feature extraction capabilities.
3) *Global Dynamic Dependency Capture:* The ST-Transformer module is applied in the spatiotemporal fusion stage to effectively capture global dynamic spatiotemporal dependencies.
4) *Comprehensive Prediction Solution:* Our model integrates global similarities of graph structures with local dynamics of temporal sequences to achieve a comprehensive traffic flow prediction solution for complex transportation systems.

II. Related Work

In recent years, traffic flow prediction has emerged as a critical research area within ITS. Driven by increasing demands for traffic management and planning, researchers have extensively explored this field, encompassing the development of traditional statistical methods, deep learning techniques, and spatiotemporal GNNs.

A. Conventional Methods

Conventional time-series forecasting methods, such as the autoregressive integrated moving average (ARIMA) model [25], vector autoregression (VAR) [26], support vector machine (SVM) [27], Kolmogorov–Arnold network (KAN) [28], and support vector regression (SVR) [29], have been commonly employed for traffic flow prediction. These models rely on mathematical and statistical frameworks, utilizing historical data under the assumption that future trends will mirror past observed patterns. Although they perform well under specific conditions, these approaches exhibit significant limitations when dealing with nonstationary and highly complex time-series data.

B. Deep Learning Technologies

With the advancement of deep learning techniques, researchers apply recurrent neural networks (RNNs) and their derivative models, including long short-term memory (LSTM) networks and gated recurrent units (GRUs). Lu et al. [30] proposed a hybrid prediction model based on RNNs, which combines the ARIMA model with LSTM networks. This model enhances accuracy by integrating the predictive abilities of both models through a dynamically weighted moving window. Shu et al. [31] introduced an improved GRU model known as the bidirectional GRU (Bi-GRU), which employs the Adam optimizer and cosine annealing learning rate for short-term traffic flow prediction to improve prediction accuracy and convergence speed. Other similar improved models include the LSTM-RNN [32], ARIMA-RNN [33], and RNN-LF [34]. These models excel in capturing the temporal correlations of traffic data due to their ability to effectively handle sequential information. However, their limitation lies in the inadequate utilization of the inherent spatial information in traffic data.

C. Spatiotemporal Methods

Convolutional neural networks (CNNs) have been employed to model spatial correlations within Euclidean spaces to effectively capture and utilize spatial information in traffic data. Chen et al. [35] proposed a hybrid neural network algorithm combining CNNs with LSTM networks. This method effectively addresses nonlinear historical data and peak-period uncertainties by leveraging the strengths of the CNN and LSTM in multivariate analysis, thereby improving the accuracy of short-term traffic flow prediction. Li et al. [36] developed a deep spatiotemporal adaptive 3-D CNN (ST-A3DNet), which employs adaptive 3-D convolution modules to simultaneously capture spatiotemporal relationships and flexibly allocate weights to historical data. The integration of an ex-mask module further considers the impact of external factors on traffic flow [36]. However, CNNs are typically limited to standard grid-like data, which poses challenges when addressing irregular traffic networks.

To overcome the limitations of CNNs in these scenarios, researchers introduce hybrid models that combine CNNs with RNNs to simultaneously capture spatiotemporal dependencies. Noteworthy contributions in this area include ConvLSTM [37], TPALSTM [38], and 3DNet [39], [40]. For instance, Li et al. [40] proposed the attention-based spatiotemporal 3-D residual neural network (AST3DRNet), which integrates 3-D residual networks with self-attention mechanisms. This model captures complex spatiotemporal correlations through the stacking of multiple 3-D residual units and 3-D convolution modules and features an innovative spatiotemporal attention module to efficiently model dynamic spatiotemporal associations. Although these models enhance the modeling capability of spatiotemporal features by integrating convolutional layers with RNN architectures, their reliance on Euclidean convolution poses limitations. This reliance may not fully align with the intrinsic geometric properties of traffic networks, leading to constraints in predictive performance.

D. Spatiotemporal GNNs

Graph convolutional networks (GCNs) mark a pivotal shift in modeling non-Euclidean spatial relationships. These models leverage graph structures to represent road networks to capture complex spatial dependencies effectively. Spatiotemporal graph convolutional networks (STGCNs) [9] and dynamic spatiotemporal graph-based CNNs (ASTGCNs) [41] with dynamic time warping (DTW) matrices for semantic understanding have demonstrated excellent capabilities in extracting spatial features.

Li and Zhu [42] proposed a spatiotemporal fusion GNN (STFGNN) for traffic flow prediction. By introducing a temporal graph and spatiotemporal fusion mechanism, the STFGNN enhances the modeling of complex spatiotemporal data, addressing the limitations of existing models in learning spatiotemporal dependencies. Yang et al. [43] proposed a spatiotemporal fusion adaptive graph network (STFAGN). This model captures latent spatiotemporal dependencies within data through the integration of convolutional layers and a novel adaptive dependency matrix via end-to-end training, supplementing incomplete information. Additionally, it employs fusion operations and fast-DTW to extract hidden spatiotemporal dependencies and temporal trends. Finally, the simple, yet effective ReZero connection is adopted as an improvement to the deep residual network, enhancing deep signal propagation and accelerating convergence.

The further development of graph learning frameworks, GMAN [44] and AGCRN [45], establishes mechanisms for dynamically learning optimal graph structures. The graph multiattention network (GMAN) employs an encoder–decoder architecture, where both the encoder and decoder consist of multiple spatiotemporal attention modules to model the impact of spatiotemporal factors on traffic conditions. The encoder encodes the input traffic features, while the decoder predicts the output sequences. Additionally, a transform attention layer is applied to mitigate error propagation across the predicted time steps [44]. Bai et al. [45] introduced the adaptive graph convolutional recurrent network (AGCRN), which strengthens the capability of GCNs by automatically capturing fine-grained spatiotemporal correlations in traffic sequences. This is achieved through the introduction of the node adaptive parameter learning (NAPL) module and the data adaptive graph generation (DAGG) module, eliminating the need for predefined spatial connection graphs and allowing the network to infer dependencies among traffic sequences.

These approaches concentrate on optimizing node embeddings and graph matrices to capture the evolving relationships in traffic data. However, there remains room for improvement in these methods' real-time responsiveness to sudden events or anomalies.

Recent research has emphasized the integration of Transformer models into spatiotemporal graph networks. For instance, STAEformer [46] utilizes attention mechanisms to parse and model long-term dependencies. These advancements pave the way for dynamic graph estimation techniques that better capture and leverage traffic networks' dynamic characteristics to enhance local and global correlation modeling. Attention mechanisms play a crucial role in improving model performance by focusing on significant information within large volumes of data. In traffic flow prediction, attention mechanisms are incorporated into models to effectively decode spatiotemporal correlations between different network nodes, providing a nuanced understanding of dynamic patterns.

The evolution from traditional statistical models to complex deep learning frameworks and spatiotemporal GNNs highlights the increasing emphasis in traffic flow prediction on dynamically capturing multidimensional spatiotemporal dependencies, including global spatial and local temporal correlations. This approach aims to develop more accurate and efficient predictive models.

III. Preliminaries

A. Problem Definition

In the multivariate traffic flow dataset (MIT), data is collected from N sensors on various urban roads, which include information such as location coordinates and road names. At each time step, typically ranging from 5 to 10 min, C-dimensional data (e.g., traffic volume, vehicle speed, and road occupancy) is obtained from all sensors. Consequently, at each time step, a vector $\mathbf{x} \in \mathbb{R}^{N\times C}$ is acquired. The entire dataset records data at each time step over a specific period. Therefore, the data for the T time steps preceding time t can be defined as

$$X^{(t-T+1:t)} = \begin{bmatrix} \begin{bmatrix} x_1^{(t-T+1)} \\ x_2^{(t-T+1)} \\ \vdots \\ x_N^{(t-T+1)} \end{bmatrix} & \begin{bmatrix} x_1^{(t-T+2)} \\ x_2^{(t-T+2)} \\ \vdots \\ x_N^{(t-T+2)} \end{bmatrix} \\ \cdots & \begin{bmatrix} x_1^{(t)} \\ x_2^{(t)} \\ \vdots \\ x_N^{(t)} \end{bmatrix} \end{bmatrix} \in \mathbb{R}^{T\times N\times C}. \quad (1)$$

Based on the historical traffic flow of T time steps $[\mathbf{X}^{(t-T+1)}, \mathbf{X}^{(t-T+2)}, \ldots, \mathbf{X}^{(t)}]$, the future traffic flow for T_p time steps, obtained from N sensors, is predicted as $\hat{\mathbf{X}}^{(t+1:t+T_p)}$. The corresponding definition is

$$\hat{\mathbf{X}}^{(t+1:t+T_p)} = F\left(\mathbf{X}^{(t-T+1:t)}, G, \theta\right) \quad (2)$$

where $\hat{\mathbf{X}}^{(t+1:t+T_p)} \in \mathbb{R}^{T_p\times N\times r}$. r is the dimension of the data predicted for N sensors over T_p future time steps. r is set to 1, since we only consider the flow rate in this work. $\mathbf{X}^{(t-T+1:t)} \in \mathbb{R}^{T\times N\times C}$ represents the input historical data over T time steps. G is the undirected distribution graph formed by the N sensors and θ is the parameter set of the MSTC-GAT corresponding to $F(\cdot)$.

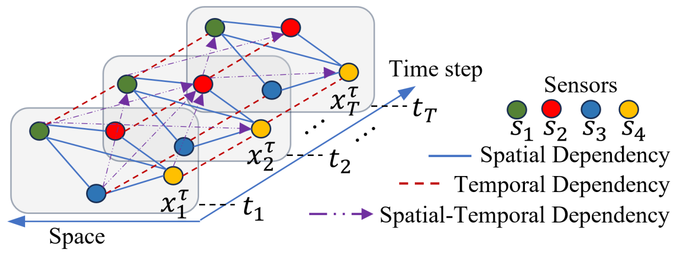


Fig. 1. Illustration of spatiotemporal dependency structure.

B. Structure of Spatial–Temporal Dependency

The spatiotemporal dependency structure in traffic flow physically reflects the structural characteristics of the traffic network and logically represents the sequential relationship over time. It also integrates the temporal and spatial connections at different times based on road and traffic flow directions at various traffic points. As illustrated in Fig. 1, which depicts the spatiotemporal dependency structure in traffic flow. There are T time steps from t_1 to t_T with data recorded by N sensors distributed across urban roads. Each time step represents a slice of data. The spatiotemporal dependency structure forms the following three main types of dependencies in the temporal and spatial dimensions.

1) Spatial Dependency: At each time step, the N sensors are distributed on roads at different geographical locations due to the connectivity between urban roads, which forms spatial dependencies between nodes. As shown in Fig. 1, the four sensors correspond to nodes $V_s = \{s_1, s_2, s_3, s_4\}$, which form an undirected graph $G_s = (V_s, \mathrm{Edge}_s)$ with edges $\mathrm{Edge}_s = \{(s_1, s_2), (s_1, s_3), (s_1, s_4), (s_2, s_4), (s_3, s_4)\}$.

2) Temporal Dependency: For each node, sequential dependencies are formed between adjacent time steps, for example, times t_1 and t_2. As illustrated, the temporal dependency graph between t_1 and t_2, $G_{(t_1\to t_2)} = (V_{(t_1\to t_2)}, \mathrm{Edge}_{(t_1\to t_2)})$, has nodes $V_{(t_1\to t_2)} = \{s_1^{(t_1)}, s_2^{(t_1)}, s_3^{(t_1)}, s_4^{(t_1)}, s_1^{(t_2)}, s_2^{(t_2)}, s_3^{(t_2)}, s_4^{(t_2)}\}$ and edges $\mathrm{Edge}_{(t_1\to t_2)} = \{(s_1^{(t_1)}, s_1^{(t_2)}), (s_2^{(t_1)}, s_2^{(t_2)}), (s_3^{(t_1)}, s_3^{(t_2)}), (s_4^{(t_1)}, s_4^{(t_2)})\}$.

3) Spatial–Temporal Dependencies: The connectivity intrinsic to urban road networks gives rise to varied effects on traffic flow at subsequent time intervals across different nodes. This results in the establishment of correlation dependencies across adjacent temporal points. Consider the dependency graph $G_{s,t_1\to t_2} = (V_{s,t_1\to t_2}, \mathrm{Edge}_{s,t_1\to t_2})$, where $V_{s,t_1\to t_2} = V_{t_1\to t_2}$. The set of edges $\mathrm{Edge}_{s,t_1\to t_2} = \{\langle s_1^{t_1}, s_2^{t_2}\rangle, \langle s_1^{t_1}, s_3^{t_2}\rangle, \langle s_2^{t_1}, s_1^{t_2}\rangle, \langle s_3^{t_1}, s_1^{t_2}\rangle, \langle s_3^{t_1}, s_2^{t_2}\rangle, \langle s_4^{t_1}, s_2^{t_2}\rangle\}$.

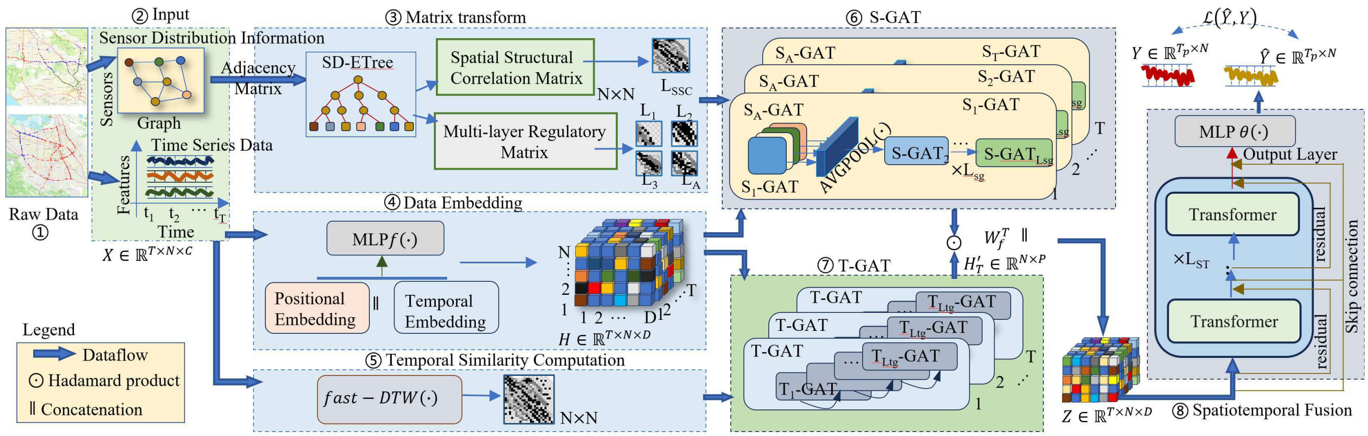


Fig. 2. Architecture of our MSTC-GAT.

In summary, spatial dependency illustrates the characteristics of the graph, temporal dependency showcases the features of sequences, and spatial–temporal dependency is a deep integration of both. To better capture these dependencies, our model needs: 1) the ability to capture the dependency features of nodes and their relationships within the graph; 2) the capability to detect the association features between different times in the sequence; and 3) the ability to embed node representations that integrate graph and sequence features to enhance the capture of spatial–temporal dependency characteristics.

IV. Methodology

A. Overall Architecture

Fig. 2 illustrates the comprehensive framework architecture of our multilayer graph attention network-based spatiotemporal correlation-aware model (MSTC-GAT).

The MSTC-GAT model fuses the matrices of hierarchical node correlation and temporal similarity between nodes by a multilayer graph attention network (GAT) to achieve spatiotemporal embedding and feature extraction. Then, we concatenate knowledge embeddings obtained from representation learning by S-GAT and T-GAT for each time step slice and feed them into the temporal attention Transformer module. We use fully connected output layer for the final prediction $\hat{Y}^{(t+1:t+T_p)} = [\hat{y}^{(t+1)}, \hat{y}^{(t+2)}, \ldots, \hat{y}^{(t+T_p)}] \in \mathbb{R}^{T_p\times N\times r}$.

As depicted in Fig. 2, the MSTC-GAT is primarily composed of the following modules: 1) graph structure correlation matrix preprocessing; 2) temporal similarity matrix computation for graph nodes; 3) multilayer spatial dependency node representation learning (S-GAT); 4) temporal similarity dependency node representation learning (T-GAT); 5) spatiotemporal fusion attention learning module (ST-Transformer); and 6) spatiotemporal fusion prediction output layer $\theta(\cdot)$.

Research has shown that the locality, globality, and correlation of space, along with the dynamics and dependencies of time and space, are crucial for modeling spatiotemporal data analysis [19], [20], [23], [24]. From the architecture of the MSTC-GAT illustrated in Fig. 2, the distribution information of sensors in the raw data is transformed through matrix operations to obtain the global distribution of roads and the interconnections among various branches, thereby providing detailed information about the spatial graph structure. During the data embedding process, the spatial positions and temporal dependencies of nodes are embedded using a multilayer perceptron (MLP), enabling the model to better understand the role of each node within the overall structure. In the temporal similarity computation section, fast dynamic time warping (fast-DTW) is employed to capture temporal dynamic correlations further, enhancing the model's interpretability. Finally, the integration challenges are addressed through the S-GAT and T-GAT blocks during spatiotemporal fusion. These modules not only capture complex dynamic patterns but also enhance the model's interpretability through robust attention mechanisms, allowing researchers to clearly understand the contribution of each module in the learning process, thus effectively addressing the inadequacies of spatial graph structures and the complexities of spatiotemporal data.

B. Graph-Structured Correlation Matrix Preprocessing Module

We formulate the spatial distribution of N sensors as a logical structural graph $G = (V, E, A)$, where $V = \{v_1, v_2, \ldots, v_N\}$ represents the set of vertices for the corresponding sensors. $E = \{(v_i, v_j)|v_i, v_j \in V, i \neq j, 1 \leq i, j \leq N\}$ denotes the set of edges existing between pairs of vertices (sensors). A is the adjacency matrix corresponding to E, where $a_{ij} \in A$ and $a_{ij} \neq 0$ (or potentially equal to 1) if there exists an edge (v_i, v_j). The sensor logical graph G can be initialized as a weighted undirected graph $G = (V, E, A)$ according to the connectivity correlations of the roads, where the adjacency matrix $A = W$ and W is the weight matrix of the edges, that is, $a_{ij} = w_{ij}$.

1) Encoding Tree Based on Minimized Structural Entropy: To capture the intrinsic spatial dependencies of data obtained from sensors, we propose an encoding tree by minimizing structural entropy to compress and encode the road sensor graphs. We denote the coding tree for graph $G = (V, E, A)$ as T, whose root node is r. $T_r = V$ represents the set of all vertices of the graph. For a nonleaf node $v \neq r$ in T, the parent node of v in T is denoted as v^-. According to the recursive definition of the tree, if T_v is a tree, $T_v^{\langle i\rangle}$ is the Ith subtree of v. In such a tree structure, each nonleaf node v corresponds to a partition of its M child nodes, represented as $T_v = \bigcup_i^M T_v^{\langle i \rangle}$, with $\bigcap_i^M T_v^{\langle i \rangle} = \emptyset$. By measuring the complexity of the graph through structural entropy and optimizing the efficiency of information encoding by minimizing this entropy, we take the contribution of vertex or edge weights to the entropy into consideration. The structural entropy for weighted graphs is thus defined as follows:

$$\begin{aligned} S_T(G) &= \sum_{v \in T, v \neq r} S_T(G; v) \\ &= -\sum_{v \in T, v \neq r} \frac{w_v}{D(G)} \log_2 \left(\frac{d_v}{d_{v^-}} \right) \end{aligned} \tag{3}$$

where $S_T(G)$ represents the structural entropy of the encoding tree T corresponding to graph G. The term w_v refers to the sum of the weights of the cut-set $[T_v, T_v/T_r]$, which is the total weight of edges connecting vertices outside T_v to all vertices within T_v. The term $D(G)$ denotes the sum of the degrees of all vertices in G, whereas d_v represents the sum of the degrees of all vertices within T_v.

2) Node Correlation Matrix: We use a greedy algorithm to minimize $S_T(G)$ to construct the encoding tree corresponding to graph G, resulting in the optimal encoding tree denoted as T^*. Utilizing the methods discussed in [47], [48], and [49], the correspondence between all vertices in G and the leaf nodes of T^* is determined. To compute the hierarchical correlation between any two leaf nodes α and β within T^*, the relative structural entropy between α and β is defined based on the structural entropy formula in (3) as follows:

$$S_{T^*}^{\text{cor}}(G; \alpha \mid \beta) = \frac{S_{T^*}^{\text{cor}}(\alpha; \beta)}{S_{T^*}^{\text{cor}}(\beta; G)} \tag{4}$$

where $S_{T^*}^{\text{cor}}(\alpha;\beta)$ represents the structural entropy between nodes α and β, and $S_{T^*}^{\text{cor}}(\beta; G)$ denotes the structural entropy of node β with respect to the entire graph G.

Based on the path between any two leaf nodes α and β within the encoding tree T^*, let γ be their nearest common ancestor node. The path-based hierarchical relative structural entropy between these nodes is further defined as follows:

$$\begin{aligned} S_{T^*}^{p\text{cor}}(G;\beta \mid \alpha) &= S_{T^*}^{\text{cor}}(G;\gamma \mid \alpha) + S_{T^*}^{\text{cor}}(G;\beta \mid \gamma) \\ &= \sum_{v, T_\alpha \subseteq T_v \subset T_\gamma} S_{T^*}^{\text{cor}}\left(G; v^- \mid v\right) \\ &\quad + \sum_{v, T_\gamma \supset T_v \supseteq T_\beta} S_{T^*}^{\text{cor}}\left(G; v \mid v^-\right) \end{aligned} \tag{5}$$

where v^- denotes the direct parent node of v within T^*.

By calculating the path-based hierarchical relative structural entropy among the N leaf nodes in T^*, we can obtain the complexity and correlated dependencies between any two of the N sensors. The path-based hierarchical relative structural entropy matrix $\mathbf{M}_{T^*}^{(H)} \in \mathbb{R}^{N \times N}$ is

$$m_{ij}^{(H)} = S_{T^*}^{p\text{cor}}\left(G; v_j \,\middle|\, v_i\right) \in \mathbf{M}_{T^*}^{(H)}, \quad i, j \in \{1, 2, \ldots, N\} \tag{6}$$

where $v_i, v_j \in T^*$. $m_{ij}^{(H)}$ represents the element at the ith row and jth column of the matrix $\mathbf{M}_{T^*}^{(H)}$, which serves as the attention between nodes v_i and v_j in the sensor distribution graph.

3) Hierarchical Attention Mask Matrix: To control the height of the encoding tree, the maximum height of T^* is set to L_{T^*}, where $4 \leq L_{T^*} \leq 6$. From the child node layers of the root node (Layer 1) of the encoding tree to the parent layers of the leaf nodes (Layer L_{T^*}), specifically from Layer 2 to Layer $L_{T^*} - 1$, each layer designates a set of nodes based on the affiliations of the leaf nodes at thecorresponding layer. We set the set of nodes at layer l to $\mathbf{v}^{(l)} = \{v_1^{(l)}, v_2^{(l)}, \ldots, v_m^{(l)}\}$, and the hierarchical attention mask matrix $\mathbf{M}_{T^*}^{(l)} = \{1, -\infty\}^{N \times N}$, where $2 \leq l \leq (L_{T^*} - 1)$

$$m_{ij}^{(l)} = \begin{cases} 1, & \text{if } \exists \gamma \in \mathbf{v}^{(l)}, v_i \in T_\gamma,\ v_j \in T_\gamma \\ -\infty, & \text{otherwise} \end{cases} \tag{7}$$

where v_i and v_j are leaf nodes. The element $m_{ij}^{(l)} \in \mathbf{M}_{T^*}^{(l)}$ corresponds to the ith row and jth column of the matrix $\mathbf{M}_{T^*}^{(l)}$, with $-\infty$ represented as `-INF`.

Consequently, we can derive the set of hierarchical attention mask matrices$\mathbf{M}_{T^*}^{(L)} = \{\mathbf{M}_{T^*}^{(2)}, \mathbf{M}_{T^*}^{(3)}, \ldots, \mathbf{M}_{T^*}^{(L_{T^*}-1)}\} \in \mathbb{R}^{N \times N}$. Here, $\mathbf{M}_{T^*}^{(1)} = \mathbf{A} \in \mathbb{R}^{N \times N}$ and $\mathbf{M}_{T^*}^{(L_{T^*})} = \mathbf{M}_{T^*}^{(H)} \in \mathbb{R}^{N \times N}$.

Ultimately, the matrix preprocessing based on the logical graph yields three types of matrices, the set of hierarchical attention mask matrices $\mathbf{M}_{T^*}^{(L)}$, the adjacency matrix of the logical graph $\mathbf{M}_{T^*}^{(1)}$, and the path-based hierarchical relative structural entropy matrix $\mathbf{M}_{T^*}^{(H)}$.

4) Temporal Similarity Computation of Nodes: To effectively capture and quantify the temporal series similarity between nodes in a traffic network, we employ the fast dynamic time warping algorithm (FAST-DTW) for multi-scale trajectory clustering (MSTC). The FAST-DTW algorithm measures the similarity between two time series to address the inherent nonlinear alignment issue for time-series data, which is often used for node similarity computation in large-scale traffic networks. In this way, our method can effectively capture temporal dependencies in traffic flow to identify similar patterns among different nodes. Thus, we enhance spatiotemporal feature fusion by considering temporal similarities between nodes. The temporal feature information is embedded with spatial features to augment the performance of the model. Moreover, our model identifies pairs of nodes with similar traffic patterns to mine the dynamic behavior of traffic networks.

Hereby, given two time series $X = (x_1, x_2, \ldots, x_p)$ and $Y = (y_1, y_2, \ldots, y_q)$, the distance matrix $\mathbf{M}_D \in \mathbb{R}^{p \times q}$ is defined as $\mathbf{M}_D(i, j) = |x_i - y_j|$, where $1 \leq i \leq p$ and $1 \leq j \leq q$. The cumulative distance matrix $\mathbf{M}_S \in \mathbb{R}^{p \times q}$ is computed according to the FAST-DTW algorithm as follows:

$$\begin{aligned} \text{DTW}(i, j) &= \mathbf{M}_S(i, j) \\ &= \mathbf{M}_D(i, j) + \min \begin{cases} \text{DTW}(i-1, j) \\ \text{DTW}(i, j-1) \\ \text{DTW}(i-1, j-1) \end{cases} \end{aligned} \tag{8}$$

where the boundary condition is given by $\text{DTW}(0, 0) = 0$. Each element $\mathbf{M}_S(i, j)$ in the matrix represents the minimum cumulative distance from the starting point of the sequence X to x_i and from the starting point of the sequence Y to y_j.

After multiple iterations of computation, $(\mathbf{M}_D(p, q))^{1/2}$ represents the similarity between the time series X and Y, that

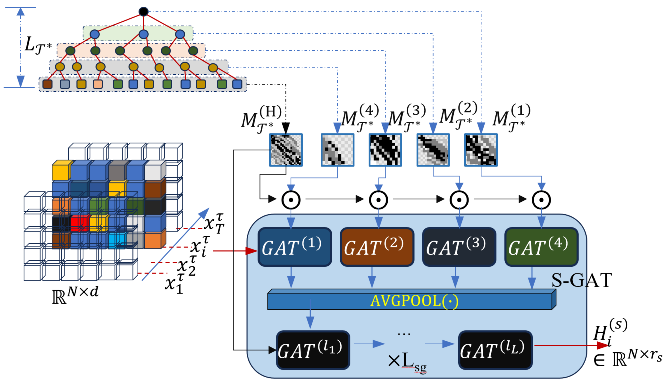


Fig. 3. Structure of the multilayer S-GAT.

is, the final distance is the square root of the element in the bottom-right corner of the cumulative cost matrix. The warping path Ω is generated by tracing back from the bottom-right corner $\mathbf{M}_D(p,q)$ to the top-left corner $\mathbf{M}_D(0,0)$

$$\Omega = (\omega_1, \omega_2, \ldots, \omega_\lambda), \quad \max(p,q) \le \lambda \le p+q \tag{9}$$

where ω_λ denotes the optimal pairing between two time series x_i and y_j. To improve the efficiency of analyzing large-scale traffic time-series data, the search is conducted over T time steps, that is, $\omega_k = (i,j)$, where $|i-j| \le T$.

For time-series data slices $Z = (z_1, z_2, \ldots, z_N)$, we use the FAST-DTW algorithm in (8) and (9) to compute the distance between the time series of each pair of nodes/sensors (i,j) following the aforementioned iterative process, where $z_i \in \mathbb{R}^{T\times r}$, with N nodes and a time step length of T. The node temporal similarity matrix $M_{\text{sim}} \in \mathbb{R}^{N\times N}$ is

$$M_{\text{sim}}(i,j) = \text{fastDTW}\left(z_i, z_j\right), \quad i,j \in \{1,2,\ldots,N\}. \tag{10}$$

C. S-GAT: Multilayer Spatial Dependency Node Representation Embedding GAT

1) Architecture of S-GAT: Fig. 3 illustrates the structure of the multilayer S-GAT. In Section IV-B, the encoding tree with a height L_{T^*} generates the matrix set $A_S = \{M_{T^*}^{(1)}, M_{T^*}^{(2)}, M_{T^*}^{(3)}, M_{T^*}^{(4)}\}$ and a path-based node correlation matrix $M_{T^*}^{(H)}$ when $L_{T^*} = 5$. At each time step t_i, multiple matrices from A_S are applied to obtain the first layer's node representation embeddings of the four GATs. These embeddings are aggregated using an average pooling layer AVGPOOL(·). Subsequently, the embeddings pass through a stacked GAT structure with L_{sg} layers to output the node representation embedding $H_i^{(s)} \in \mathbb{R}^{N\times r_s}$ for time step t_i, where r_s denotes the output feature dimension of the S-GAT.

2) Single Hybrid Adjacency Matrix-Processing GAT Layer: The input adjacency matrix is set to $A_s^{(\text{in})} = A_s^{(i)} \odot M_{T^*}^{(H)}$ for the spatial dependency GAT layer at time step t_i, where $\odot$ denotes the element-wise product operation, and $A_s^{(i)} \in A_S$. The input feature matrix is $H_S^{(0)} = x_i^\tau \in \mathbb{R}^{N\times d}$. For the lth layer, each attention head k defines the computation of attention as follows:

$$e_{ij}^{(l,k)} = \text{LeakyReLU}\left(\left(\mathbf{a}^{(l,k)}\right)^T \cdot \left[W^{(l,k)}h_i^{(l)} \,\|\, W^{(l,k)}h_j^{(l)}\right]\right) \cdot a_{ij}^{(\text{in})} \tag{11}$$

where $(\mathbf{a}^{(l,k)})^T \in \mathbb{R}^{1\times 2r_s}$ represents the learnable self-attention matrix for each node pair in the lth layer and kth head. $W^{(l,k)} \in \mathbb{R}^{2r_s}$ is the learnable linear transformation matrix. The vectors $h_i^{(l)}, h_j^{(l)} \in \mathbb{R}^{r_s}$ denote the features of nodes i and j, respectively. If the element $a_{ij}^{(\text{in})} \in A_s^{(\text{in})}$ is nonzero, there is a connection between nodes i and j. $a_{ij}^{(\text{in})}$ represents the weight of this connection, which signifies the mutual influence between the nodes.

After obtaining the attention between node pairs, the attention is normalized using softmax(·)

$$\alpha_{ij}^{(l,k)} = \frac{\exp\left(e_{ij}^{(l,k)}\right)}{\sum_{p\in\mathcal{N}(i)} \exp\left(e_{\text{ip}}^{(l,k)}\right)} \tag{12}$$

where $\mathcal{N}(i)$ denotes the set of nodes adjacent to node i with nonzero weights, as derived from the input adjacency matrix $A_s^{(\text{in})}$.

The multihead features are aggregated by normalized attention

$$h_i^{(l+1)} = \Big\|_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(l,k)} W^{(l,k)} h_j^{(l)}\right) \tag{13}$$

where $h_i^{(l+1)}$ represents the output feature of the node i. This feature is obtained by performing a weighted sum of the features of its neighboring nodes using the normalized attention weights. The nonlinear activation function $\sigma(\cdot)$ is implemented using the ReLU function.

For each stacked layer of the hybrid adjacency matrix, $L = 2/4$ layers of GAT are processed. The matrices GAT$^{(i)}$ $(i = 1,2,3,4)$ in Fig. 3 are defined as $A_{s_k}^{(\text{in})} = M_{T^*}^{(k)} \odot M_{T^*}^{(H)}$ $(k = 1,2,3,4)$.

3) Stacked Spatiotemporal Dependency GAT Layers: As shown in Fig. 3, each of the $(L_{T^*} - 1)$ individual hybrid adjacency matrix-processed GAT layers, GAT$^{(i)}$ $(L_{T^*} = 5, i = 1,2,3,4)$, are subjected to mean aggregation through the average pooling layer AVGPOOL(·)

$$H_i^{(s_1)} = \frac{1}{L_{T^*} - 1} \sum_{l=1}^{L_{T^*}-1} H_i^{\left(A_{s_l}^{(\text{in})}\right)} \tag{14}$$

where L_{T^*} denotes the height of the encoding tree. This height is the parent node layers from the root node to the leaf nodes. Each layer corresponds to an attention mask matrix. $H_i^{(A_{s_l}^{(\text{in})})}$ represents the feature embedding obtained by processing the temporal data slice at time step t_i, where $A_{s_l}^{(\text{in})}$ is the adjacency matrix through the corresponding GAT layer.

The obtained $H_i^{(s_1)}$ is used as the input feature. The node correlation matrix $M_{T^*}^{(H)}$ is used as the input adjacency matrix. We use these two matrices to form a series of L_{sg} $(L_{sg} = 2)$ GAT layers in sequence

$$H_i^{(s_2)} = \text{GAT}^{(l_1)}\left(H_i^{(s_1)}, M_{T^*}^{(H)}\right) \tag{15}$$

$$H_i^{(S)} = H_i^{(s_3)} = \text{GAT}^{(L_{sg})}\left(H_i^{(s_2)}, M_{T^*}^{(H)}\right) \tag{16}$$

where $H_i^{(S)} \in \mathbb{R}^{N\times r_s}$ represents the feature output obtained by the entire S-GAT from the temporal data slice at time step t_i. We concat L_{sg} $(L_{sg} = 2)$ GAT layers to preserve the feature

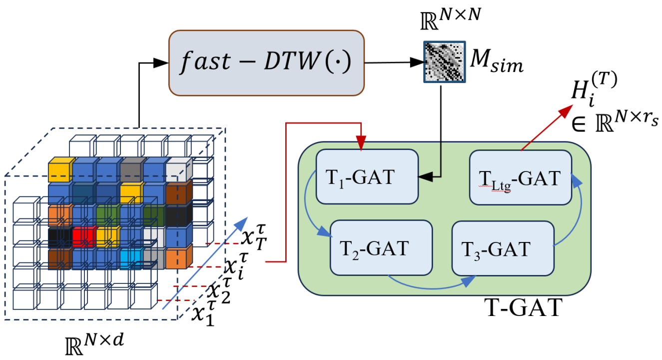


Fig. 4. Structure of multilayer T-GAT.

of node correlation ($M_{T^*}^{(H)}$) for higher-level feature representations. Thus, this way enhances the model's expressive and generalization capabilities.

D. T-GAT: Multilayer Temporal Dependency Node Representation Embedding GAT

As illustrated in Fig. 4, we design a temporal dependency graph attention network (T-GAT) structure to handle temporal data. The input at time step t_i is a temporal data slice corresponding to the feature matrix $H_T^{(0)} = x_i^\tau \in \mathbb{R}^{N\times d}$. The node temporal similarity matrix $M_{\text{sim}} \in \mathbb{R}^{N\times N}$ is obtained using fast-DTW(·) from the temporal step T (sample temporal data), which serves as the adjacency matrix input to T-GAT. The T-GAT is composed of L_{tg} ($L_{tg} = 4$) layers of k-head attention GATs in series. Each layer T_i-GAT is defined as follows:

$$H_i^{(T_{l+1})} = \text{GAT}^{(T_{l+1})}\left(H_i^{(T_l)}, M_{\text{sim}}\right), \quad l = \{0, 1, 2, 3\} \tag{17}$$

where $\text{GAT}^{(T_{l+1})}$ refers to (11)–(13).

The output of the final layer $H_i^{(T)} = H_i^{(T_{l+1})} \in \mathbb{R}^{N\times r_s}$ represents the temporally dependent node feature representation obtained after processing through all T-GAT layers.

E. Weighted Spatiotemporal Fusion Attention Learning

As illustrated in Fig. 2, the node features for each of the T time steps, denoted as $H^{(S)}$ and $H^{(T)} \in \mathbb{R}^{N\times r_s}$, are obtained from the S-GAT and T-GAT, respectively. To learn the optimal combination of spatial and temporal features during training, we employ a learnable parameter matrix for weighted fusion. For each time step t and node i, the weighted fusion can be expressed as

$$H_{t,i}^{(ST)} = \theta\left(W^{ST}\left(H_{t,i}^{(S)} \odot H_{t,i}^{(T)}\right)\right) \tag{18}$$

where $H_{t,i}^{(S)}, H_{t,i}^{(T)} \in \mathbb{R}^{r_s}$ represent the S-GAT feature $H^{(S)}$ and the T-GAT feature $H^{(T)}$ for node i at time step t, respectively. $\odot$ denotes the element-wise product (Hadamard product), which is used for spatial and temporal feature fusion. $\theta(\cdot)$ is the activation function ReLU. The overall fused feature matrix $H^{(ST)} \in \mathbb{R}^{T\times N\times r_s}$ is computed by (18).

F. Transformer-Based Spatiotemporal Fusion Model

As shown in the right part of Fig. 2, the Transformer prediction module consists of L_{tr} stacked Transformer layers with residual connections. Each layer includes a multihead self-attention mechanism and a feedforward neural network (FNN). They further encode the outputs of the spatiotemporal fusion from S-GAT and T-GAT to capture spatiotemporal interactions across multiple projection spaces.

We perform attention computation simultaneously over both temporal and sensor dimensions to capture complex spatiotemporal dependencies across time steps and sensors. To achieve this, we reshape the input $H^{(ST)} \in \mathbb{R}^{T\times N\times r_s}$ by merging the first two dimensions, T and N, into a single dimension. We merge T and N of $H^{(ST)}$ to obtain the feature matrix $Z^{(0)}$, such that $Z^{(0)} \in \mathbb{R}^{(T\times N)\times r_s}$. $T \times N$ denotes the combined dimension of time steps and sensors, with each combination corresponding to a feature vector.

Assume that there are L_{tr} layers in total. For the ith Transformer layer ($1 \le \text{Tr}_i \le L_{\text{tr}}$), the hth head attention has learnable matrices $W_q^{(\text{Tr}_i,h)}$, $W_k^{(\text{Tr}_i,h)}$, and $W_v^{(\text{Tr}_i,h)}$. The queries, keys, and values in H projection spaces are given by

$$\begin{aligned}
\text{MultiHead}\left(Z^{(\text{Tr}_i)}\right) &= W^{(\text{Tr}_i)}\text{Concat}\left(\text{head}_1^{(\text{Tr}_i)}, \ldots, \text{head}_h^{(\text{Tr}_i)}\right)\\
\text{head}_h^{(\text{Tr}_i)} &= \text{Attention}\left(Q_h^{(\text{Tr}_i)}, K_h^{(\text{Tr}_i)}, V_h^{(\text{Tr}_i)}\right)\\
&= V_h^{(\text{Tr}_i)}\,\text{softmax}\left(\frac{\left(K_h^{(\text{Tr}_i)}\right)^T Q_h^{(\text{Tr}_i)}}{\sqrt{d_k}}\right)\\
Q_h^{(\text{Tr}_i)} &= W_q^{(\text{Tr}_i,h)} Z^{(\text{Tr}_i)}\\
K_h^{(\text{Tr}_i)} &= W_k^{(\text{Tr}_i,h)} Z^{(\text{Tr}_i)}\\
V_h^{(\text{Tr}_i)} &= W_v^{(\text{Tr}_i,h)} Z^{(\text{Tr}_i)} \quad \forall n \in \{1, 2, \ldots, N\}
\end{aligned} \tag{19}$$

where N denotes the number of attention heads.

The size of the column vectors in the input matrices $Q_h^{(\text{Tr}_i)}$ and $K_h^{(\text{Tr}_i)}$ is denoted as d_k. Similarly, d_v is the dimension of the column vectors in $V_h^{(\text{Tr}_i)}$. The weight matrix for the output of the multihead attention is denoted as $W_0 \in \mathbb{R}^{r_s\times \text{N}d_v}$. The matrices $W_q^{(\text{Tr}_i,n)} \in \mathbb{R}^{r_s\times d_k}$, $W_k^{(\text{Tr}_i,n)} \in \mathbb{R}^{r_s\times d_k}$, and $W_v^{(\text{Tr}_i,n)} \in \mathbb{R}^{r_s\times d_v}$ are the weight matrices for the query, key, and value transformations, respectively.

The computation for the residual connection and layer normalization within the layer is expressed as follows:

$$Z' = \text{LayerNorm}\left(Z^{(\text{Tr}_i)} + \text{MultiHead}\left(Z^{(\text{Tr}_i)}\right)\right). \tag{20}$$

The FNN is defined as follows:

$$\text{FNN}\left(Z'\right) = W_{2f} \cdot \text{ReLU}\left(W_{1f} \cdot Z' + b_{1f}\right) + b_{2f} \tag{21}$$

where Z' is the vector at each position in the input sequence from the previous layer. The matrices W_{1f} and W_{2f} are the weight matrices of the two-layer neural network, and b_{1f} and b_{2f} are the corresponding biases. All these are learnable network parameters.

The Transformer architecture with multilayer residual connections and layer normalization is

$$Z^{(\text{Tr}_{i+1})} = \text{LayerNorm}\left(Z' + \text{FNN}\left(Z'\right)\right) \tag{22}$$

where Z' and $\text{FNN}(Z')$ are computed as per (19) and (20), respectively. $Z^{(\text{Tr}_{i+1})} \in \mathbb{R}^{T\times N\times r_s}$.

TABLE I
SUMMARY OF THE TRAFFIC FLOW PUBLIC DATASETS

Datasets	#Nodes	#TimeSteps	#Edges	Time Range
PeMS04	307	16,992	340	1/1/2018–2/28/2018
PeMS08	170	17,856	295	7/1/2016–8/31/2016
METR-LA	207	34,272	407	3/1/2012–6/30/2012
PEMS-BAY	325	52,116	827	1/1/2017–5/31/2017

G. Output of the Prediction Model and Loss Function

The feature matrix obtained by spatiotemporal fusion over T time steps is used as the input for the prediction model to forecast a single data feature of N sensors over future T_p time steps. The output of the spatiotemporal fusion model is $H^{(\mathrm{STTr})} \in \mathbb{R}^{T\times N\times r_s}$ and the predicted output is $\hat{Y} \in \mathbb{R}^{T_p\times N\times 1}$. To obtain the prediction output, the reconstructed encoding is used as the input to the final MLP layer for training

$$\hat{y} = \mathrm{MLP}\left(H^{(\mathrm{STTr})}\right) = H^{(\mathrm{STTr})}W_{\mathrm{out}} + b_{\mathrm{out}} \tag{23}$$

where $W_{\mathrm{out}} \in \mathbb{R}^{r_s\times 1}$ is a learnable weight matrix and b_{out} is the bias term.

To learn the parameters within the network, the prediction model employs the Huber loss function to leverage stochastic gradient descent for iterative learning. The Huber loss function exhibits robustness when handling prediction errors, which balances the squared error and the absolute error performance for small and large errors. The Huber loss function of the model can be defined as

$$L_\delta\left(Y^{(t+1:t+T_p)}, \hat{Y}^{(t+1:t+T_p)}\right) = \frac{1}{T_p\times N}\sum_{t=1}^{T_p}\sum_{i=1}^{N}\mathrm{Er}_\delta\left(Y,\hat{Y}\right) \tag{24}$$

$$\mathrm{Er}_\delta\left(Y,\hat{Y}\right) = \begin{cases}\frac{1}{2}\left(Y-\hat{Y}\right)^2, & \text{if } \left|Y-\hat{Y}\right| \le \delta\\ \delta\cdot\left(\left|Y-\hat{Y}\right| - \frac{1}{2}\delta\right), & \text{otherwise}\end{cases} \tag{25}$$

where δ is a hyperparameter controlling the sensitivity of the squared error. $Y^{(t+1:t+T_p)}$ and $\hat{Y}^{(t+1:t+T_p)}$ represent the ground truth and predictions of traffic flow over future T_p time steps, respectively. The hyperparameter δ behaves as a threshold in the Huber loss function to determine the application of squared loss or absolute loss to achieve a robust loss.

The MSTC model effectively fuses the spatiotemporal features through multiple layers of GNNs and a Transformer for accurately forecasting data in future time steps. The residual connections and layer normalization facilitate the stable training of deeper networks to ensure the model's adaptability and predictive capability on large datasets.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Configuration and Description

1) Datasets: To validate the effectiveness and superiority of the multilayer graph attention network-based spatiotemporal correlation-aware model (MSTC-GAT), we utilized four publicly available traffic flow datasets, that is, PeMS04, PeMS08, METR-LA, and PEMS-BAY. The fundamental characteristics of these datasets are detailed below, with the corresponding statistical results presented in Table I.

Each time step is regarded as an instance, and the sample proportions for training, testing, and validation are set as 7:2:1 based on the total length of time steps in each dataset.

The initialization of the data representation embedding layer can be defined as

$$H_{\mathrm{en}} = \mathrm{MLP}(X) = XW_{\mathrm{en}} + b_{\mathrm{en}} \tag{26}$$

where $H_{\mathrm{en}} \in \mathbb{R}^{T\times N\times D}$, with D being the dimension after representation embedding. W_{en} and b_{en} are the weight matrix and bias for node representation embedding, respectively.

2) Evaluation Metrics: In the analysis of the experimental results, the following three primary performance evaluation metrics were employed: 1) mean absolute error (MAE); 2) root-mean-squared error (RMSE); and 3) mean absolute percentage error (MAPE)

$$\mathrm{MAE} = \frac{1}{N_S}\sum_{i=1}^{N_S}\left|Y_i - \hat{Y}_i\right| \tag{27}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_S}\sum_{i=1}^{N_S}\left(Y_i - \hat{Y}_i\right)^2} \tag{28}$$

$$\mathrm{MAPE} = \frac{1}{N_S}\sum_{i=1}^{N_S}\left|\frac{Y_i - \hat{Y}_i}{Y_i}\right| \times 100\% \tag{29}$$

where N_S denotes the total number of observable samples; Y_i represents the true outcome of the ith sample; $\hat{Y}_i$ signifies the predicted outcome of the ith sample by the model.

3) Baseline Methods: To evaluate the performance of the proposed MSTC-GAT in comparison with other models, this study includes classical and state-of-the-art traffic flow prediction models in the comparative analysis.

For the METR-LA and PEMS-BAY datasets, we compare our MSTC-GAT with 10 state-of-the-art models, that is, FC-LSTM [50], STGCN [9], ASTGCN [41], STID [51], DCRNN [10], GWNet [52], STAEformer [46], GL-STGTN [8], DDSTGCN [12], AGCRN [45], and DSTET [53].

For the PeMS04 and PeMS08 datasets, our comparison excludes STAEformer [46] and GL-STGTN [8], and adds GMAN [44] and MTGNN [54], since they have not conducted on these two datasets.

4) MSTC-GAT Model Parameter Configuration: In this study, the experimental setup is as follows: the environment utilizes a CPU (Xeon[1] Platinum 8255C), a GPU (NVIDIA RTX V100 with 32 GB memory), and is operated on Ubuntu. The development environment employs Python 3.8 with the deep learning framework PyTorch 1.9.0 and TorchVision 0.10.0, with GPU training acceleration supported by CUDA 11.0.

The initial training configuration of the MSTC-GAT model is detailed in Table II.

B. Experimental Results Analysis

Experiments were conducted using the datasets from Table I and the model configurations specified in Table II. The MSTC-GAT model was compared with the classical FC-LSTM model and several state-of-the-art prediction models based on GNNs

[1]Registered trademark.

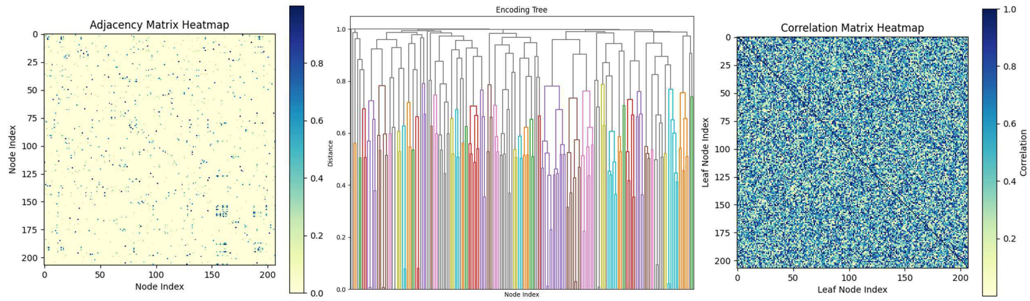


Fig. 5. Heatmaps of the adjacency matrix, coding tree based on minimal structural entropy, and node correlation matrix for the METR-LA dataset.

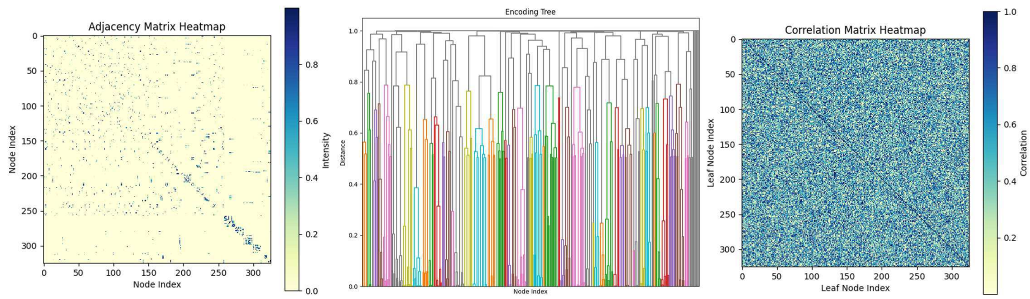


Fig. 6. Heatmaps of the adjacency matrix, coding tree based on minimal structural entropy, and node correlation matrix for the PEMS-BAY dataset.

TABLE II

MODEL PARAMETER CONFIGURATIONS

Parameter	Description	Value
L_{T^*}	Height of the encoding tree	5
T	Input time steps	12
T_p	Output/prediction time steps	3/6/12
D	Node representation embedding dimension	64/128
H	Number of heads in attention mechanism	8
L_{tr}	Number of GAT layers in the S-GAT module	4
L_{sg}	Number of stacked GAT layers in the S-GAT module	2
L_{tg}	Number of stacked layers in the T-GAT module	4
l_r	Learning rate of the model	0.001–0.01
δ	Sensitivity control of squared error	[0.9, 1]
Bsize	The batch size	16/32

and Transformer architectures. Figs. 5 and 6 present the heatmaps of the original adjacency lists, the coding trees based on minimal structural entropy, and the node correlation matrices derived from the coding trees for the METR-LA and PEMS-BAY datasets.

The experimental results are presented in Table III. Table III shows the prediction results for 15, 30, and 60 min using four benchmark datasets, that is, METR-LA, METR-LA, PEMS04, and PEMS08 datasets.

The analysis of the data presented in Table III highlights the superior performance of MSTC-GAT across all tested temporal intervals for the METR-LA dataset. Specifically, for the 15-min forecast, MSTC-GAT achieved an MAE of 2.42. The approximate error reduction is from 10% to 15% compared to other models such as STGCN (MAE = 2.88) and DCRNN (MAE = 2.77). Our MSTC-GAT also recorded a 0.04 decrease in MAE compared to the subsequent best-performing model, DSTET, further establishing its efficacy in short-term prediction tasks. Furthermore, MSTC-GAT's MAPE of 8.18 in the 60-min forecast was significantly lower than that of all competing models, underscoring its robust capability in learning spatial–temporal characteristics over extended horizons.

In the PEMS-BAY dataset evaluation, the MAE of MSTC-GAT for the 15-min forecast is 1.08, slightly above the optimal DSTET model (MAE = 1.06). It demonstrates exceptional performance in terms of RMSE and MAPE, recording values of 2.15 and 2.11, respectively. This underscores MSTC-GAT's enduring robustness and stability under complex data conditions. Moreover, its prediction errors remained at a minimal level for the 30- and 60-min intervals, highlighting the model's comprehensive advantages in mid-to-long-term forecasting tasks.

The superiority of MSTC-GAT is particularly pronounced in the PEMS04 dataset. It achieved an MAE of 16.53 for the 15-min forecast, significantly outperforming models like DDSTGCN (MAE = 17.69) and DCRNN (MAE = 18.53). Compared to DSTET, MSTC-GAT also realized slight improvements in RMSE and MAPE metrics, further demonstrating its adaptability in handling various traffic flow patterns.

Finally, for the PEMS08 dataset, predictions across all timeframes showcased MSTC-GAT's leading edge, particularly in long-duration (60-min) forecasts where the MAE was only 13.79, outperforming the current best model, DSTET (MAE = 14.11).

Through comprehensive performance comparisons, the MSTC-GAT model demonstrates a highly flexible and accurate approach to traffic forecasting by integrating multilevel spatial–temporal contexts and self-attention mechanisms. Its accuracy and stability across various datasets and temporal scales surpass those of existing mainstream models. This indicates that MSTC-GAT not only excels in understanding and learning complex traffic flow patterns but also possesses

TABLE III

RESULTS FOR 15/30/60 MIN PREDICTION ON THE METR-LA, METR-LA, PEMS04, AND PEMS08 DATASETS

Dataset	Method	15 min			30 min			60 min		
		MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
METR-LA	FC-LSTM	3.44	6.30	9.60	3.77	7.23	10.9	4.37	8.96	13.2
	STGCN	2.88	5.74	7.62	3.47	7.24	9.57	4.59	9.40	12.7
	ASTGCN	2.67	5.19	6.83	3.03	6.10	8.34	3.49	7.25	9.79
	STID	2.80	5.53	7.70	3.18	6.60	9.40	3.55	7.54	10.95
	DCRNN	2.77	5.38	7.30	3.15	6.45	8.80	3.60	7.60	10.5
	GWNet	2.69	5.15	6.90	3.07	6.22	8.37	3.53	7.37	10.01
	STAEformer	2.65	5.11	6.85	2.97	6.00	8.13	3.34	7.02	9.70
	GL-STGTN	2.61	5.08	6.94	2.96	5.97	8.23	3.37	6.99	9.81
	DDSTGCN	2.64	5.01	6.76	3.00	6.02	8.12	3.44	7.13	9.74
	AGCRN	2.86	5.55	7.55	3.25	6.57	8.99	3.67	7.56	10.46
	DSTET	2.46	4.52	6.05	2.71	5.22	6.95	3.03	6.08	8.30
	MSTC-GAT	2.42	4.44	6.09	2.65	5.17	6.87	2.98	6.02	8.18
PEMS-BAY	FC-LSTM	2.05	4.19	4.8	2.2	4.55	5.2	2.37	4.96	5.7
	STGCN	1.36	2.96	2.9	1.81	4.27	4.17	2.49	5.69	5.79
	ASTGCN	1.3	2.74	2.79	1.61	3.65	3.64	1.88	4.1	4.42
	STID	1.3	2.81	2.73	1.62	3.72	3.68	1.89	4.4	4.47
	DCRNN	1.31	2.76	2.73	1.65	3.75	3.71	1.97	4.6	4.68
	GWNet	1.3	2.74	2.73	1.63	3.7	3.67	1.95	4.52	4.63
	STAEformer	1.31	2.78	2.76	1.62	3.68	3.62	1.88	4.34	4.41
	GL-STGTN	1.26	2.73	2.79	1.55	3.58	3.63	1.8	4.3	4.38
	DDSTGCN	1.29	2.71	2.69	1.6	3.64	3.62	1.89	4.37	4.46
	AGCRN	1.37	2.93	2.95	1.7	3.89	3.88	1.99	4.64	4.72
	DSTET	1.06	2.17	2.13	1.29	2.85	2.74	1.57	3.6	3.54
	MSTC-GAT	1.08	2.15	2.11	1.25	2.81	2.68	1.51	3.53	3.41
PEMS04	FC-LSTM	21.37	33.31	15.21	23.72	36.58	18.02	26.76	40.28	20.94
	STGCN	18.74	29.84	14.42	19.64	31.34	13.27	21.12	33.53	14.22
	ASTGCN	19.68	31.23	13.23	21.45	33.74	14.54	25.57	39.23	17.34
	STID	17.51	28.48	12	18.29	29.86	14.46	19.58	31.79	13.38
	DCRNN	18.53	29.61	12.71	19.65	31.37	13.45	21.67	34.19	15.03
	GWNet	18	28.83	13.64	18.96	30.33	14.23	20.53	32.54	15.41
	GMAN	18.27	29.35	12.66	18.81	30.85	13.25	20.01	31.32	13.4
	MTGNN	18.65	30.13	13.32	19.48	32.02	14.08	20.96	34.46	14.96
	DDSTGCN	17.69	28.38	12.34	18.51	29.54	12.7	19.51	30.69	13.46
	AGCRN	18.52	29.79	12.31	19.45	31.45	12.82	20.64	33.31	13.74
	DSTET	16.9	27.38	11.59	17.68	28.66	12.44	18.48	29.77	12.71
	MSTC-GAT	16.53	27.22	11.38	17.43	27.86	12.11	17.81	28.23	12.32
PEMS08	FC-LSTM	17.38	26.27	12.63	21.22	31.97	17.32	30.69	43.96	25.72
	STGCN	14.95	23.48	9.87	15.92	25.36	10.42	17.65	28.03	11.34
	ASTGCN	16.4	25.27	9.91	18.41	28.23	10.99	22.41	33.7	13.34
	STID	13.28	21.66	8.62	14.21	23.57	9.24	15.58	25.89	10.33
	DCRNN	14.16	22.2	9.31	15.24	24.26	9.9	17.7	27.14	11.13
	GWNet	13.72	21.71	8.8	14.67	23.5	9.49	16.15	25.95	10.74
	GMAN	13.8	22.88	9.41	14.62	24.12	9.57	15.72	26.47	10.56
	MTGNN	14.3	22.55	10.56	15.25	24.41	10.54	16.8	26.96	10.9
	DDSTGCN	13.54	21.14	8.82	14.02	22.23	9.71	14.7	24.12	10.41
	AGCRN	14.51	22.87	9.34	15.66	25	10.34	17.49	27.93	11.72
	DSTET	12.25	20.24	8.03	13.21	21.63	8.83	14.11	23.53	9.3
	MSTC-GAT	12.08	19.75	7.74	12.79	20.88	8.52	13.79	23.37	9.12

strong generalization capabilities to handle diverse urban traffic networks.

An in-depth analysis of Table III and Fig. 7 reveals the performance enhancements achieved by the MSTC-GAT model across various datasets and provides a detailed comparison to other models. In the METR-LA dataset, MSTC-GAT achieves improvements of 29.65%, 29.52%, and 36.56% in MAE, RMSE, and MAPE for the 15-min prediction compared to the FC-LSTM model, respectively. This underscores the model's capability to capture short-term traffic dynamics with greater accuracy. During the 30-min forecast, the MSTC-GAT surpasses STGCN by 23.63%, 28.59%, and 28.21% in MAE, RMSE, and MAPE, respectively. It highlights the proficiency in medium-term dynamic feature extraction. For the 60-min prediction, MSTC-GAT exhibits improvements of 14.61%, 16.97%, and 16.45% in MAE, RMSE, and MAPE over DCRNN, respectively. These results reflect the model's significant long-term spatial–temporal sequence modeling capabilities.

The PEMS-BAY dataset further accentuated MSTC-GAT's comparative advantages. In the 15-min prediction, the model achieves 16.92%, 21.53%, and 24.37% improvements in MAE, RMSE, and MAPE over ASTGCN, respectively. The 30-min prediction shows a 30.94% enhancement in MAE and a 34.19% improvement in RMSE relative to STID, demonstrating the model's superior mid-term prediction capability. For the 60-min forecast, a 22.68% improvement in MAPE is noted against GWNet, indicating MSTC-GAT's adeptness at capturing data trends over extended periods.

On the PEMS04 dataset for the 60-min prediction, MSTC-GAT outperforms AGCRN with an 8.71% improvement in MAE and an 8.02% enhancement in RMSE, showcasing its excellence in large-scale traffic network analysis.

Finally, analysis of the PEMS08 dataset shows that for the 15-min forecast, MSTC-GAT improves MAE and MAPE by 30.49% and 38.71%, respectively. During the 30-min prediction, the model's MAE and RMSE improvements are 39.72% and 34.69% against FC-LSTM, emphasizing its

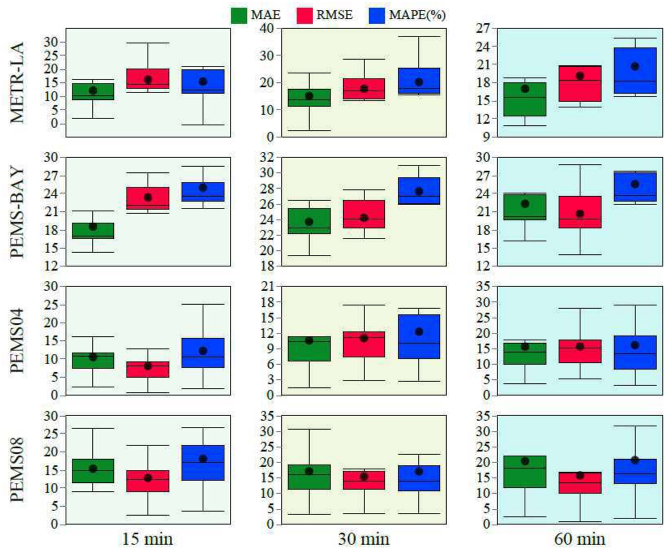


Fig. 7. Boxplots of performance improvements of MSTC-GAT compared to other models across four datasets.

accuracy in varied temporal scenarios. For the 60-min prediction, improvements are 21.15% in MAE and 22.18% in MAPE compared to other methodologies.

Across all datasets and timeframes, MSTC-GAT consistently demonstrates superiority. In short-term predictions (e.g., 15 min), the model adjusts to predict instantaneous traffic fluctuations accurately. In medium-term forecasts (e.g., 30 min), it maintains high accuracy. In long-term predictions (60 min), its performance remains robust. These results highlight MSTC-GAT's universal capability and adaptability across different temporal scales.

The MSTC-GAT's multidimensional ability to capture and analyze the intrinsic nature of traffic flows through temporal and spatial dependencies significantly reduces cumulative errors via GNNs and time-series processing mechanisms, achieving comprehensive and accurate traffic forecasts.

Through the comparisons made in the aforementioned experiments, the MSTC-GAT model efficiently integrates temporal and spatial information for traffic flow prediction. The computational cost primarily stems from the number of sensors and the complexity of the map. Specifically, the matrix transformations of sensor distribution and the construction of the adjacency matrix contribute significantly to the computational burden. Additionally, the MLP and the fast-DTW modules also require substantial computation when processing spatial and temporal embeddings. In terms of model interpretability, the use of S-GATs and T-GATs enhances the understanding of each component. Despite the relatively high computational costs, comparisons with existing models, such as the LSTM and GCN, demonstrate that our model offers notable advantages in capturing complex temporal and spatial relationships.

C. Ablation Study

To verify the necessity and effectiveness of each component within the MSTC-GAT model for enhancing overall performance, an ablation study was conducted on the METR-LA and PEMS-BAY datasets. The study focused on the following modules: 1) the node correlation matrix module based on the minimal structure entropy encoding tree (MST); 2) the fast-DTW-based node temporal similarity matrix (FDM); 3) the multilayer spatiotemporal dependency representation learning gat module (MGAT); and 4) the spatial–temporal fusion Transformer module (STTformer). Four corresponding ablation experiments are designed as follows.

1) *MSTC-GAT w/o MST:* This configuration removes the node correlation matrix module based on the MST. Without this MST module, the basic adjacency matrix formed by the spatial configuration of the N sensors in the original dataset is directly used as input for the spatial dependency GAT layer (S-GAT). The GNN is then employed to capture propagation features among nodes.
2) *MSTC-GAT w/o FDM:* This setting excludes the fast-DTW-based node temporal similarity matrix and replaces it with temporal attention between nodes based on time series. An attention matrix between nodes is obtained and used as an input for the temporal dependency GAT layer (T-GAT) to capture temporal dependency propagation features through GAT.
3) *MSTC-GAT w/o MGAT:* This setup modifies the structure used in S-GAT and T-GAT, which leverages multilayer GAT propagation learning to capture deep propagation features of spatiotemporal dependencies, by employing a single-layer GAT model instead.
4) *MSTC-GAT w/o STTformer:* This variant omits the STTformer, utilizing the output from an MLP fully connected layer directly for prediction.

The results of these experiments are presented in Table IV.

The experimental results presented in Table IV allow for further analysis of the impact on model performance following the removal of the respective modules, as depicted in Fig. 8.

In the comparative analysis of the MSTC-GAT model and its submodels, synthesizing the data from Table IV and the curves depicted in Fig. 8, it is evident that the proposed MSTC-GAT outperforms the other incomplete submodels in both efficiency and effectiveness. Specifically, on the METR-LA dataset, MSTC-GAT achieves MAE values of 2.42, 2.65, and 2.98 for 15-, 30-, and 60-min forecasts, respectively. All are notably lower than those of the submodels. Among these submodels, MSTC-GAT w/o STTformer exhibits the poorest performance, with MAE values of 3.11, 3.92, and 5.22 at the same intervals. According to the curve in Fig. 8, the MAE of MSTC-GAT for 15-, 30-, and 60-min predictions is lower by 22%, 32%, and 43% compared to MSTC-GAT w/o STTformer, respectively. This significant difference is also reflected in the RMSE and MAPE metrics, further underscoring the critical role of the STTformer module in capturing spatiotemporal dynamics.

A similar trend is observed in the PEMS08 dataset. Similarly, the MAE for MSTC-GAT is 29.4%, 28.2%, and 34.4% lower for the 15-, 30-, and 60-min predictions compared to the poorest performing MSTC-GAT w/o STTformer, respectively. These results demonstrate that the STTformer module is indispensable for capturing spatiotemporal features in both long- and short-term time series.

Additionally, observations indicate that the MSTC-GAT w/o FDM submodel, which excludes the FDM (multimodal feature

TABLE IV
ABLATION STUDY RESULTS ON THE METR-LA AND PEMS08 DATASETS

Dataset	Method	15 min			30 min			60 min		
		MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
METR-LA	MSTC-GAT w/o MST	2.92	5.97	7.71	3.63	7.35	10.63	5.02	9.52	13.29
	MSTC-GAT w/o FDM	3.05	6.04	7.92	3.86	7.61	10.84	5.16	9.55	13.44
	MSTC-GAT w/o MGAT	2.83	6.01	7.64	3.57	7.28	10.48	4.93	9.37	13.11
	MSTC-GAT w/o STTformer	3.11	6.08	8.02	3.92	7.69	10.91	5.22	9.61	13.65
	MSTC-GAT	**2.42**	**4.44**	**6.09**	**2.65**	**5.17**	**6.87**	**2.98**	**6.02**	**8.18**
PEMS08	MSTC-GAT w/o MST	15.12	23.92	10.19	16.41	26.31	11.38	18.27	29.24	11.95
	MSTC-GAT w/o FDM	16.07	24.18	10.32	17.09	26.89	12.15	19.33	31.52	13.17
	MSTC-GAT w/o MGAT	15.06	23.35	9.72	15.66	25.14	11.07	17.13	28.02	11.03
	MSTC-GAT w/o STTformer	17.14	25.33	11.09	17.81	27.61	13.02	21.05	33.20	14.27
	MSTC-GAT	**12.08**	**19.75**	**7.74**	**12.79**	**20.88**	**8.52**	**13.79**	**23.37**	**9.12**

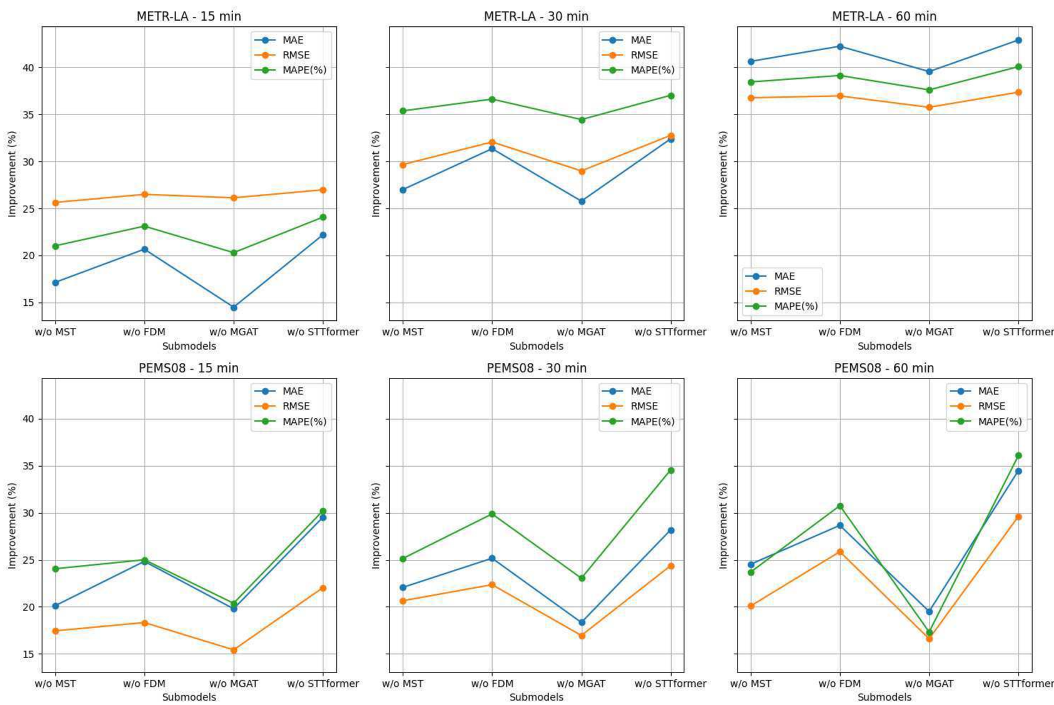


Fig. 8. Comparison of performance improvements of MSTC-GAT over other submodels in ablation experiments on the METR-LA and PEMS08 datasets.

extraction) module, also performs relatively poorly across all metrics. Within the METR-LA dataset, removing this module leads to increases in MAE by 26%, 46%, and 73% for the 15-, 30-, and 60-min forecasts, highlighting the importance of FDM in MSTC-GAT, respectively. In summary, the MSTC-GAT achieves efficient handling of complex traffic flowthrough its complete module configuration, significantly enhancing prediction accuracy while demonstrating excellent adaptability and robustness.

VI. CONCLUSION

Traffic flow prediction has an increasing emphasis on capturing the dynamic, global spatial, and local temporal correlations associated with multidimensional spatiotemporal dependencies. To address the challenges posed by these dependencies, global similarity, and local dynamics, we propose an MSTC-GAT for traffic flow prediction. The model's multilayer S-GAT employs hierarchical attention masks and path-based node correlation matrices as inputs to its S-GAT to capture both local and global spatial dependencies in traffic data. The temporal structure-aware module (T-GAT) constructs a node short-term similarity matrix based on time slices as input to the T-GAT to capture local dynamic temporal dependencies. Finally, the ST-Transformer learns from the weighted spatiotemporal feature-fused node embeddings to capture global dynamic dependencies, thus producing the prediction outcomes. Extensive experiments are conducted on four public datasets compared with 10 state-of-the-art traffic flow prediction models.

Experimental results demonstrate that short-term prediction (15 min) showed average performance improvements of 14.09%, 15.04%, and 17.64%, respectively. The improvements for medium-term prediction (30 min) are 16.59%, 17.05%, and 19.25%, while that for long-term prediction (60 min) are 18.79%, 17.78%, and 20.74%. These results demonstrate that the MSTC-GAT consistently outperforms other models. Additionally, ablation experiments on the METR-LA and PEMS-BAY datasets highlight MSTC-GAT's remarkable adaptability and robustness. In the future, we will explore the model's node spatiotemporal feature embeddings by contrastive learning to further improve prediction accuracy.

References

[1] T. Garg and G. Kaur, "A systematic review on intelligent transport systems," *J. Comput. Cognit. Eng.*, vol. 2, no. 3, pp. 175–188, Jun. 2022.

[2] W. Fang, X. Li, Z. Lin, J. Zhou, and T. Zhou, "Mixture correntropy with variable center LSTM network for traffic flow forecasting," *Digit. Transp. Saf.*, vol. 3, no. 4, pp. 264–270, 2024.

[3] J. Zhu, W. Bai, J. Zhao, L. Zuo, T. Zhou, and K. Li, "Variational mode decomposition and sample entropy optimization based transformer framework for cloud resource load prediction," *Knowl.-Based Syst.*, vol. 280, Nov. 2023, Art. no. 111042.

[4] W. Bai, J. Zhu, J. Zhao, W. Cai, and K. Li, "An unsupervised multidimensional representation learning model for short-term electrical load forecasting," *Symmetry*, vol. 14, no. 10, p. 1999, Sep. 2022.

[5] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatiotemporal networks based on multitask deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 468–478, Mar. 2020.

[6] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial–temporal graph data for traffic forecasting," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5415–5428, Nov. 2022.

[7] H. Yao et al., "Deep multi-view spatial–temporal network for taxi demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 2588–2595.

[8] Z. Li, J. Zhou, Z. Lin, and T. Zhou, "Dynamic spatial aware graph transformer for spatiotemporal traffic flow forecasting," *Knowl.-Based Syst.*, vol. 297, Aug. 2024, Art. no. 111946.

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

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

[11] H. Zhang, Z. Lin, H. Xie, J. Zhou, Y. Song, and T. Zhou, "Two-way heterogeneity model for dynamic spatiotemporal traffic flow prediction," *Knowledge-Based Syst.*, vol. 320, Jun. 2025, Art. no. 113635.

[12] Y. Sun et al., "Dual dynamic spatial–temporal graph convolution network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 23680–23693, Dec. 2022.

[13] D. Cai, K. Chen, Z. Lin, D. Li, T. Zhou, and M.-F. Leung, "JointSTNet: Joint pre-training for spatial–temporal traffic forecasting," *IEEE Trans. Consum. Electron.*, vol. 71, no. 2, pp. 6239–6252, May 2025.

[14] X. Shu, L. Zhang, G.-J. Qi, W. Liu, and J. Tang, "Spatiotemporal co-attention recurrent neural networks for human-skeleton motion prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3300–3315, Jun. 2022.

[15] C. Zheng et al., "Node classification in temporal graphs through stochastic sparsification and temporal structural convolution," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, Sep. 2021, pp. 330–346.

[16] J. Tang, X. Shu, R. Yan, and L. Zhang, "Coherence constrained graph LSTM for group activity recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 636–647, Feb. 2022.

[17] X. Shu, B. Xu, L. Zhang, and J. Tang, "Multi-granularity anchor-contrastive representation learning for semi-supervised skeleton-based action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7559–7576, Jun. 2023.

[18] D. Xu, J. Liang, W. Cheng, H. Wei, H. Chen, and X. Zhang, "Transformer-style relational reasoning with dynamic memory updating for temporal network modeling," in *Proc. Nat. Conf. Artif. Intell.*, vol. 35, Sep. 2021, pp. 4546–4554.

[19] F. Wang, G. Zhu, H. Ding, P. Zhang, C. Yuan, and Y. Huang, "Boosting temporal graph learning from perspectives of global and local structures," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 8, pp. 15359–15373, Aug. 2025.

[20] G. Liang, P. Tiwari, S. Nowaczyk, S. Byttner, and F. Alonso-Fernandez, "Dynamic causal explanation based diffusion-variational graph neural network for spatiotemporal forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 5, pp. 9524–9537, May 2025.

[21] R. Yan, L. Xie, J. Tang, X. Shu, and Q. Tian, "HiGCIN: Hierarchical graph-based cross inference network for group activity recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 6955–6968, Jun. 2023.

[22] D. Xu, W. Cheng, D. Luo, X. Liu, and X. Zhang, "Spatio-temporal attentive RNN for node classification in temporal attributed graphs," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3947–3953.

[23] T. Zhang, W. Jiao, J. Yu, and Y. Xiong, "Short-term residential load forecasting framework based on spatial–temporal fusion adaptive gated graph convolution networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 9, pp. 17004–17018, Sep. 2025.

[24] Z. Wu, D. Zheng, S. Pan, Q. Gan, G. Long, and G. Karypis, "TraverseNet: Unifying space and time in message passing for traffic forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 2003–2013, Feb. 2024.

[25] Q. Shi et al., "Block Hankel tensor ARIMA for multiple short time series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, pp. 5758–5766.

[26] J. M. B. Haslbeck, L. F. Bringmann, and L. J. Waldorp, "A tutorial on estimating time-varying vector autoregressive models," *Multivariate Behav. Res.*, vol. 56, no. 1, pp. 120–149, Jan. 2021.

[27] A. Roy and S. Chakraborty, "Support vector machine in structural reliability analysis: A review," *Rel. Eng. Syst. Saf.*, vol. 233, May 2023, Art. no. 109126.

[28] Z. Lin, D. Wang, C. Cao, H. Xie, T. Zhou, and C. Cao, "GSA-KAN: A hybrid model for short-term traffic forecasting," *Mathematics*, vol. 13, no. 7, p. 1158, Mar. 2025.

[29] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.

[30] S. Lu, Q. Zhang, G. Chen, and D. Seng, "A combined method for short-term traffic flow prediction based on recurrent neural network," *Alexandria Eng. J.*, vol. 60, no. 1, pp. 87–94, Feb. 2021.

[31] W. Shu, K. Cai, and N. N. Xiong, "A short-term traffic flow prediction model based on an improved gate recurrent unit neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16654–16665, Sep. 2022.

[32] A. A. Kashyap et al., "Traffic flow prediction models—A review of deep learning techniques," *Cogent Eng.*, vol. 9, no. 1, 2022, Art. no. 2010510.

[33] V. Rajalakshmi and S. G. Vaidyanathan, "Hybrid time-series forecasting models for traffic flow prediction," *Promet-Traffic Transp.*, vol. 34, no. 4, pp. 537–549, Jul. 2022.

[34] A. Belhadi, Y. Djenouri, D. Djenouri, and J. C.-W. Lin, "A recurrent neural network for urban long-term traffic flow forecasting," *Int. J. Speech Technol.*, vol. 50, no. 10, pp. 3252–3265, Oct. 2020.

[35] K. Chen et al., "Dynamic spatio-temporal graph-based CNNs for traffic flow prediction," *IEEE Access*, vol. 8, pp. 185136–185145, 2020.

[36] H. Li, X. Li, L. Su, D. Jin, J. Huang, and D. Huang, "Deep spatio-temporal adaptive 3D convolutional neural networks for traffic flow prediction," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 2, pp. 1–21, Apr. 2022.

[37] R. He, Y. Liu, Y. Xiao, X. Lu, and S. Zhang, "Deep spatio-temporal 3D densenet with multiscale ConvLSTM-resnet network for citywide traffic flow forecasting," *Knowledge-Based Syst.*, vol. 250, Aug. 2022, Art. no. 109054.

[38] S.-Y. Shih, F.-K. Sun, and H.-Y. Lee, "Temporal pattern attention for multivariate time series forecasting," *Mach. Learn.*, vol. 108, nos. 8–9, pp. 1421–1441, Sep. 2019.

[39] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan, "Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3913–3926, Oct. 2019.

[40] L. Li, F. Dai, B. Huang, S. Wang, W. Dou, and X. Fu, "AST3DRNet: Attention-based spatio-temporal 3D residual neural networks for traffic congestion prediction," *Sensors*, vol. 24, no. 4, p. 1261, Feb. 2024.

[41] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 922–929.

[42] M. Li and Z. Zhu, "Spatial–temporal fusion graph neural networks for traffic flow forecasting," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 4189–4196.

[43] S. Yang, H. Li, Y. Luo, J. Li, Y. Song, and T. Zhou, "Spatiotemporal adaptive fusion graph network for short-term traffic flow forecasting," *Mathematics*, vol. 10, no. 9, p. 1594, May 2022.

[44] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 1, pp. 1234–1241.

[45] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 17804–17815.

[46] H. Liu et al., "Spatio-temporal adaptive embedding makes vanilla transformer SOTA for traffic forecasting," in *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2023, pp. 4125–4129.

[47] A. Li and Y. Pan, "Structural information and dynamical complexity of networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3290–3339, Jun. 2016.

[48] A. Li et al., "Decoding topologically associating domains with ultra-low resolution Hi-C data by graph structural entropy," *Nature Commun.*, vol. 9, no. 1, p. 3265, Aug. 2018.

[49] D. Zou et al., "MultiSPANS: A multi-range spatial–temporal transformer network for traffic forecast via structural entropy optimization," in *Proc. 17th ACM Int. Conf. Web Search Data Mining*, Mar. 2024, pp. 1032–1041.

[50] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.

[51] Z. Shao, Z. Zhang, F. Wang, W. Wei, and Y. Xu, "Spatial–temporal identity: A simple yet effective baseline for multivariate time series forecasting," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2022, pp. 4454–4458.

[52] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial–temporal synchronous graph convolutional networks: A new framework for spatial–temporal network data forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 914–921.

[53] W. Sun, R. Cheng, Y. Jiao, J. Gao, Z. Zheng, and N. Lu, "Transformer network with decoupled spatial–temporal embedding for traffic flow forecasting," *Int. J. Speech Technol.*, vol. 53, no. 24, pp. 30148–30168, Dec. 2023.

[54] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 753–763.

Junjie Liu received the Ph.D. degree from Aberystwyth University, Aberystwyth, U.K., in 2014.

He is currently an Associate Professor with Zhaoqing University, Zhaoqing, China. His main research fields include deep learning and its applications (such as graph neural networks), 3-D image processing, machine vision, pattern recognition, and machine learning.

Yu Wang received the B.S. degree in computer science and engineering from South China University of Technology (SCUT), Guangzhou, China, in 2005, and the M.S. degree in computer science from China University of Petroleum (East China), Qingdao, China, in 2018.

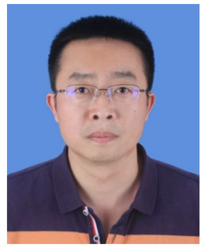
He is with the School of Computer Science, Guangdong University of Petrochemical Technology, Maoming, China. His research interests include cloud edge computing and scheduling optimization.

Jiaxian Zhu received the M.Sc. degree in computer science from Guangdong University of Technology, Guangzhou, China, in 2006.

She is currently an Associate Professor with the School of Computer Science and Software Engineering, Zhaoqing University, Zhaoqing, China. Her current research interests include scheduling optimization, large-scale resource scheduling, and deep learning and its applications.

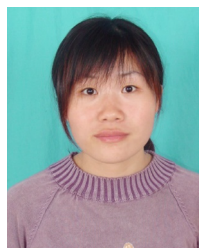
Weihua Bai received the Ph.D. degree in computer science and engineering from South China University of Technology (SCUT), Guangzhou, China, in 2017.

He is currently a Professor with the School of Computer Science and Software Engineering, Zhaoqing University, Zhaoqing, China. His research interests include cloud computing, scheduling optimization, large-scale resource scheduling, and deep learning and its applications.



Huibing Zhang received the B.S. and M.S. degrees in computer science and technology from Guilin University of Electronic Technology, Guilin, China, in 2000 and 2005, respectively, and the Ph.D. degree in computer science and technology from Beijing University of Technology, Beijing, China, in 2012.

He is currently a Professor with Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology. He presided over and participated in a number of NSFC projects. His research interests include educational big data, the Internet of Things, and AI.



Liyun Zuo received the B.S. degree in engineering and management from Zhengzhou University, Zhengzhou, China, in 2003, the M.S. degree in computer science from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2006, and the Ph.D. degree in computer science and engineering from South China University of Technology (SCUT), Guangzhou, China, in 2016.

She is a Professor with Guangdong University of Petrochemical Technology, Maoming, China. Her research interests include cloud edge computing and scheduling optimization.

Teng Zhou (Member, IEEE) received the M.Eng. degree from Sun Yat-sen University (SYSU), Guangzhou, China, in 2012, under the supervision of Prof. Xiaonan Luo, and the Ph.D. degree from South China University of Technology (SCUT), Guoqzhou, in 2017, under the supervision of Prof. Guoqiang Han.

He was a Post-Doctoral Research Fellow with the Center of Smart Health, The Hong Kong Polytechnic University, Hong Kong. He is currently a Full Professor with the School of Cyberspace Security (School of Cryptology), Hainan University, Haikou, China. His research interests include spatiotemporal data analysis, traffic flow theory, and graph convolutional networks.

Keqin Li (Fellow, IEEE) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1985, and the Ph.D. degree in computer science from the University of Houston, Houston, TX, USA, in 1990.

He is a SUNY Distinguished Professor at the State University of New York, New Paltz, NY, USA, and a National Distinguished Professor at Hunan University, Changsha, China.

Dr. Li is a member of the SUNY Distinguished Academy. He is an AAAS Fellow, an AAIA Fellow, an ACIS Fellow, and an AIIA Fellow. He is a member of the European Academy of Sciences and Arts. He is a member of the Academia Europaea (Academician of the Academy of Europe).