ADHSTGCN: Adaptive Dual Hypergraph Spatiotemporal Convolutional Network for Traffic Prediction

Tong Zhang, ZhiFeng Wang*

Abstract—**Spatio-temporal graph modeling is essential for achieving accurate traffic flow prediction and enhancing the efficiency of transportation systems. Traditional models often rely on static, predefined graphs that fail to fully capture the complexity of real-world road networks. Although global semantic adjacency matrices have been introduced to enrich proximity-based graphs and facilitate association modeling in graph convolutional networks (GCNs), they overlook the dynamic spatiotemporal characteristics of traffic systems, thereby limiting their effectiveness in capturing nuanced dependencies. Moreover, many existing spatio-temporal graph neural networks (STGNNs) exhibit suboptimal performance during end-to-end training due to their reliance on fixed graph structures. To address this limitation, we propose a novel framework named Adaptive Dual Hypergraph Spatio-Temporal Graph Convolutional Network (ADHSTGCN).**

In this framework, we design a Dual Hypergraph Convolutional Network (DHGC) as the core prediction module. The DHGC integrates edge features into the learning process, thereby enhancing the representation of edge-level interactions and capturing more intricate spatiotemporal dependencies. Additionally, inspired by the Expectation-Maximization (EM) algorithm, we introduce an Adaptive Collaborative Graph Learning (ACGL) module. By alternately optimizing the parameters of the DHGC and ACGL modules, each module can iteratively refine its estimates based on the latest updates from the other module, leading to improved convergence and stability.

To validate the effectiveness of the proposed ADHSTGCN model, extensive experiments were conducted on four public traffic datasets. Compared with the best baseline models, ADHSTGCN achieves average improvements of 1.87% in MAPE for speed prediction tasks (METR-LA, PEMS-BAY) and 4.45% for flow prediction tasks (PEMS03, PEMS04). These results highlight the model's superior accuracy in both fine-grained and long-term forecasting, as well as its capability to capture complex spatiotemporal dependencies in urban traffic systems.

Index Terms—**Traffic prediction, expectation maximization, hypergraph, graph convolution.**

I. Introduction

WITH the rapid growth of urban populations and vehicles, transportation systems have become increasingly complex and face numerous challenges such as congestion. Consequently, the development of modern intelligent transportation systems (ITS) for cities has become imperative. In this context, traffic prediction, as a crucial task in ITS, has garnered widespread attention and emerged as a research hotspot. Accurately capturing the evolving spatiotemporal patterns within large-scale transportation networks constitutes the fundamental difficulty in traffic forecasting. Throughout the development of this field, researchers have progressively established multiple methodological frameworks. Initial solutions predominantly employed linear analytical techniques, exemplified by Historical Average (HA) [1] and Moving Average (MA) [2] algorithms that generate traffic projections through systematic aggregation of past observations. The methodological evolution subsequently witnessed the emergence of sophisticated statistical paradigms, particularly clustering-based KNN techniques [3], [4], enhanced ARIMA formulations [5],[6] that integrate Autoregressive (AR) and MA components with differencing operations to handle non-stationary data, along with multivariate VAR systems [7] extending traditional AR models. Although these methodologies demonstrate computational efficiency and straightforward implementation, their capability to model complex traffic interactions remains constrained, ultimately limiting predictive performance enhancements. Later approaches with better generalization capabilities and smaller errors were proposed, including Support Vector Regression (SVR) [9], Random Forest Regression (RFR) [11], Fuzzy Logic Regression (FLR) [12], Kalman Filters (KFs)[8], as a variant of Support Vector Machines (SVM) [10] and and their variants such as Extended Kalman Filters (EKFs)[8], and hybrid models [13]. However, their performance remained suboptimal. Compared to these traditional methods, neural network-based approaches have gained prominence due to their superior ability to capture nonlinear relationships in traffic data. Examples include neural networks [14], [15], Bayesian networks [16], and Recurrent Neural Networks (RNNs) [17]. With advancements in deep learning [18], numerous deep learning-based traffic prediction methods [19] have emerged, achieving state-of-the-art performance by leveraging deep neural networks to capture complex data features. Temporal neural networks like LSTM [20], [21] and GRU [22] are commonly used to learn temporal features, but sequential learning models face challenges in training complexity and computational costs. For spatial feature extraction, CNN-based methods like ST-ResNet [23] were introduced, but these approaches treat road networks as grid-like images, losing their inherent graph structure. Recently, Graph Convolutional Networks (GCNs) [24] and Temporal Convolutional Networks (TCNs) [25] have been integrated into traffic prediction. GCNs preserve the graph structure of transportation networks, while TCNs capture temporal dynamics. Their fusion has led to high-performance GCN-based

Manuscript received April 17, 2025; revised August 14, 2025.

Tong Zhang is a postgraduate student of the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China (e-mail: zt963927783@gmail.com).

ZhiFeng Wang* is an Associate Professor at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China (Corresponding author: phone: +086-150-4234-1839; e-mail: wangzhifeng_sia@126.com).

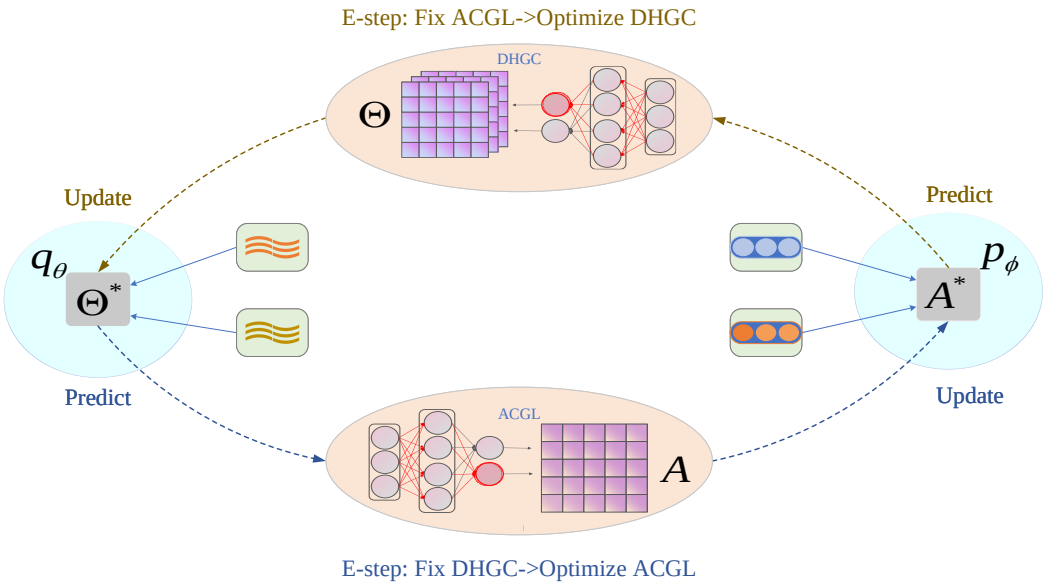


Fig. 1: Simple framework based on em algorithms.

frameworks such as STGCN [26], Graph WaveNet [27], and DGCNN [28]. These methods involve constructing graphs or adjacency matrices where road segments (nodes) and their relationships (edges) are explicitly defined. Early approaches focused on creating graphs based solely on direct connectivity between road segments [29]. To enhance precision, subsequent studies incorporated distance-based [30] and geographic similarity-based [31] relationships. However, static graphs fail to adequately capture the dynamic spatiotemporal characteristics of traffic flow; for example, real-world events such as accidents or congestion can significantly alter traffic correlations between road segments. Recent advancements address this limitation by leveraging real-time traffic data to construct dynamic graphs [32], [33], thereby achieving substantial improvements over static models [34].

Despite the insights provided by dynamic graph convolutional network (GCN)-based methods in modeling traffic flow dynamics, several challenges remain unresolved. Specifically, edges, which represent relationships between nodes, lack directly observable features beyond traffic direction and structural connectivity, unlike nodes that possess measurable traffic data. Consequently, enriching edge feature representation emerges as a critical issue. Moreover, contemporary end-to-end frameworks exhibit inherent architectural limitations, where predictive accuracy is constrained by the adequacy of graph representation learning. Additionally, the mutually dependent optimization of co-adaptive modules originates from suboptimal initialization states, leading to coupled optimization dynamics that propagate initial parameter biases throughout the training process. More critically, the joint optimization framework lacks explicit mechanisms for directional regularization, enabling uncontrolled error propagation across interconnected network components during gradient-based parameter updates.

To tackle these challenges, we propose ADHSTGCN, an Adaptive Dual Hypergraph Spatiotemporal Graph Convolutional Network based on the Expectation-Maximization (EM) algorithm. As illustrated in Fig. 1, inspired by the EM framework, our approach divides the traffic prediction task into two key components: the Dual Hypergraph Convolutional (DHGC) prediction module and the Adaptive Collaborative Graph Learning (ACGL) module. The DHGC module integrates traffic flow graphs with dual hypergraph associations to more effectively capture edge features. Meanwhile, the ACGL module optimizes graph structures via a carefully designed loss function while ensuring sparsity. Instead of traditional end-to-end training, we adopt an alternating optimization strategy. In the DHGC training phase, a predefined graph from the graph set is treated as the optimal structural expectation and remains fixed. During the ACGL training phase, the optimized DHGC module is frozen to generate more effective adjacency matrices. These two processes iteratively alternate, with a sparsity-enforcing loss function ensuring that the generated graphs maintain conciseness. This approach not only refines spatial dependencies but also uncovers hidden node correlations, thereby enhancing prediction accuracy and providing significant value for traffic management and control. The main contributions of this paper are summarized as follows:

1) Addressing the Limited Expressive Power of Predefined Adjacency Matrix Graphs by Incorporating Edge Features through Dual Hypergraphs: To overcome the limitations of predefined adjacency matrix graphs, this paper proposes leveraging dual hypergraphs to incorporate edge features into the model learning process.

2) Proposing the DHGC Network Module to Enhance the Adaptability of Graph Learning Models: To improve the adaptability of graph learning models in capturing urban traffic flow characteristics, this study introduces the DHGC module. Within this network, both the predefined graph and its dual hypergraph are fully utilized to effectively exploit edge features for traffic flow prediction.

3) Developing an Adaptive Dual Hypergraph Neural Network Framework Based on the EM Algorithm: This paper presents the ACGL module and an innovative modeling framework that employs a stepwise iterative training method. Through two alternating steps, the framework progressively enhances

the model's explanatory power over the data.

II. RELATED WORK

A. Traffic Prediction Based on a Single Time Series Feature

In the field of traffic forecasting, utilizing single-variable time series features has been a traditional methodology for modeling and prediction. This approach involves an in-depth analysis of time series data to identify key temporal patterns such as trends, seasonality, and cyclic behaviors, enabling a better understanding of passenger flow dynamics. Among classical statistical models, the Autoregressive Integrated Moving Average (ARIMA) model is particularly recognized for its efficiency in handling stationary time series. Li et al. [33] validated the effectiveness of ARIMA in modeling passenger flow at Sanya Airport, highlighting its practical applicability [34]. Furthermore, Ding et al. [35] enhanced conventional forecasting techniques by integrating Generalized Autoregressive Conditional Heteroskedasticity (GARCH) with ARIMA, which effectively improved the accuracy of traffic flow predictions [36]. The empirical validation of ARIMA's efficacy in transportation modeling was first demonstrated through Li et al.'s seminal study on passenger flow dynamics at Sanya Airport, establishing its operational robustness in real-world scenarios. Building upon this foundation, Ding et al. [37] achieved superior forecasting precision through systematic integration of Generalized Autoregressive Conditional Heteroskedasticity (GARCH) with ARIMA frameworks, marking a critical methodological innovation in handling traffic flow volatility. However, the fundamental constraints of ARIMA in addressing three critical challenges – nonlinear temporal fluctuations, non-stationary data distributions, and multidimensional temporal patterns – became increasingly apparent with expanding dataset complexity and escalating precision requirements. This recognition precipitated a methodological shift toward deep learning architectures, particularly recurrent neural networks (RNNs) equipped with specialized memory gating mechanisms. Among these innovations, Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) emerged as paradigm-shifting solutions due to their architectural capacity to model both long-range temporal dependencies and nonlinear interaction mechanisms. The evolutionary trajectory witnessed significant milestones with the development of Temporal-Domain Enhanced LSTM (T-LSTM) [37], a temporal feature preservation framework for univariate traffic flow forecasting that demonstrated 12% accuracy improvement through intrinsic data encoding. Concurrently, the SVR-LSTM hybrid architecture proposed by Guo et al. [38] effectively captured traffic flow anomalies by synergistically combining kernel-based pattern recognition with sequential modeling. To further refine the modeling of temporal dependencies in traffic forecasting, Zhao et al. adopted the Temporal Convolutional Network (TCN) model. This architecture, based on a one-dimensional convolutional neural network (1D-CNN), effectively captures long-range sequential dependencies [39]. In 2020, Sha et al. implemented a GRU-based forecasting system capable of predicting time intervals ranging from 15 minutes to 6 hours, significantly improving traffic management, safety monitoring, and passenger flow regulation [40]. More recently, in 2023, VN. Katambie et al. introduced a hybrid framework that fuses LSTM with ARIMA, combining the strengths of statistical modeling and deep learning-based sequence prediction. This integration has substantially enhanced the reliability and robustness of traffic flow forecasting, offering a more precise depiction of traffic dynamics [41].

B. GCN-based traffic prediction

Graph Convolutional Networks (GCNs) (Kipf Welling, 2017) have demonstrated strong capabilities in capturing spatial dependencies within non-Euclidean domains, leading to significant advancements in areas such as protein structure analysis and social network modeling. In recent years, GCN-based spatiotemporal neural networks have gained increasing attention in traffic forecasting. Li et al proposed the Diffusion Convolutional Recurrent Neural Network (DCRNN), which integrates graph convolution operations into a recurrent framework based on diffusion processes to enhance average speed prediction. Building upon this, researchers introduced the Attention-Based Spatial-Temporal Graph Convolutional Network (ASTGCN), which leverages both spatial and temporal attention mechanisms to refine predictive accuracy. To further optimize model performance, Li and Zhanxing (2021) incorporated a specialized temporal graph into the framework.

Despite these advancements, the inherent uncertainty in node relationships presents challenges in accurately capturing spatial dependencies. To address this limitation, recent studies have explored graph learning modules to dynamically infer graph structures. Wu et al. proposed Graph WaveNet, which constructs an adaptive dependency matrix to model bidirectional sparse interactions between nodes. However, this approach struggles to enforce strict sparsity constraints on the learned adjacency matrix. Subsequently, Wu et al. developed an enhanced graph learning layer designed to extract unidirectional dependencies while incorporating sparsity constraints. Nevertheless, empirical results indicate that this method remains suboptimal for end-to-end training frameworks. Guo et al. introduced a Laplacian matrix learning approach that dynamically constructs graph structures based on input data, alleviating inaccuracies in predefined graphs. However, this method still faces limitations in effectively capturing latent node correlations.

C. Spatio-temporal Correlation-Based traffic prediction

Current spatiotemporal graph networks primarily evolve along two technical pathways: RNN-based and CNN-based methods. Early RNN-based approaches modeled spatiotemporal dependencies by filtering input data and passing features extracted through graph convolution to recurrent units. Subsequent studies introduced various optimization strategies, such as enhancing information propagation through diffusion convolution or employing attention mechanisms to strengthen focus on critical features [42]. Additional research proposed node-level RNNs and edge-level RNNs to separately model different aspects of temporal sequence information [43]. However, RNN architectures exhibit lower computational efficiency when processing long sequences and are more prone to gradient explosion when combined with GCNs.

In contrast, CNN-based methods adopt architectures combining graph convolution with one-dimensional convolutional neural networks (1D-CNNs) to improve computational efficiency [43]. Although these methods offer computational advantages over RNNs in terms of complexity, expanding the model's receptive field typically requires stacking multiple convolutional layers or incorporating global pooling mechanisms to capture long-range dependencies.

III. PRELIMINARIES

A. Problem Formulation

The topological structure of transportation networks can be formally characterized through a weighted digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where urban road segments constitute the vertex set $\mathcal{V}$ ($|\mathcal{V}| = N$), and spatiotemporal correlations between segment traffic states define the edge set $\mathcal{E}$. As visualized in Fig. 2(a), this relational structure is encoded in the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$, whose elements quantify connection weights derived from either road network connectivity or inter-segment spatial proximities.

The dynamic traffic measurements (speed/flow/occupancy) across the network at timestamp t form a feature tensor $\mathbf{X}_t = (x_t^1, x_t^2, \ldots, x_t^N)^\top \in \mathbb{R}^{N\times F}$, where each nodal observation $x_t^i \in \mathbb{R}^F$ contains F-dimensional attributes. Over a temporal window $[1, \tau]$, the evolving network states constitute a spatiotemporal sequence:

$$\mathcal{X}^{1:\tau} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_\tau)^\top \in \mathbb{R}^{N\times F\times \tau}$$

Formally, the traffic forecasting task requires learning a mapping function $\mathcal{F}$ that projects historical observations $\mathcal{X}^{(t-T'+1):t}$ and graph structure $\mathcal{G}$ onto future states $\mathcal{X}^{(t+1):(t+T)}$:

$$\mathcal{F}\left(\mathcal{X}^{(t-T'+1):t}, \mathcal{G}\right) \to \mathcal{X}^{(t+1):(t+T)} \tag{1}$$

where $\mathcal{X}^{(t-T'+1):t} \in \mathbb{R}^{N\times F\times T'}$ encodes T'-step historical patterns and $\mathcal{X}^{(t+1):(t+T)} \in \mathbb{R}^{N\times F\times T}$ denotes T-step predictions.

B. Graph and Hypergraph Convolution Networks

Graph Convolutional Networks (GCNs) are a type of convolutional network designed for graphs, where node features are updated by aggregating and transforming the features of neighboring nodes. Kipf *et al.* [11] introduced a first-order Chebyshev polynomial approximation for GCN, formulated as follows: The first-order approximation of Graph Convolutional Networks (GCNs) is defined as a layer-wise transformation function:

$$\mathrm{GCN}(X) = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}XW\right) \tag{2}$$

where $\tilde{A} = A + I_N \in \mathbb{R}^{N\times N}$ is the adjacency matrix with self-connections, $X \in \mathbb{R}^{N\times F}$ denotes the input node feature matrix containing F-dimensional features for N nodes, and $W \in \mathbb{R}^{F\times F'}$ represents the trainable weight matrix. The symmetric normalized matrix $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ stabilizes gradient propagation by controlling feature magnitudes, balances node influence through degree normalization, and preserves directional consistency via self-loop integration.

The complete architecture is constructed by successively applying the GCN layer:

$$X^{(l+1)} = \mathrm{GCN}(X^{(l)}) \tag{3}$$

where $X^{(l)}$ indicates the hidden node representations at layer l. The nonlinear activation function $\sigma(\cdot)$ (ReLU) enables model capacity enhancement between layers. The simplified formula is as follows:

$$\mathrm{GCN}(\mathbf{X}) = \hat{A}\mathbf{X}\Theta \tag{4}$$

where $\hat{A} \in \mathbb{R}^{N\times N}$ is the normalized adjacency matrix, $\mathbf{X} \in \mathbb{R}^{N\times F_1}$ is the input feature matrix, and $\Theta \in \mathbb{R}^{F_1\times F_2}$ is the parameter to learn.

The hypergraph convolution network (HGCN) can be regarded as the extension of the GCN from graph to hypergraph. Feng *et al.* proposed a hypergraph convolution as follows:

$$\mathrm{HGCN}(\mathbf{X}_h) = D_{ho}^{-\frac{1}{2}}HWD_{he}^{-\frac{1}{2}}H^\top D_{ho}^{-\frac{1}{2}}\mathbf{X}_h\Theta \tag{5}$$

where $D_{he} \in \mathbb{R}^{N\times N}$ and $D_{ho} \in \mathbb{R}^{E\times E}$ denote the diagonal matrices of the hyper-edge degrees and the hyper-node degrees, respectively. Here, we call the node of hypergraph as hyper-node for distinguishing from that of the general graph. $H \in \mathbb{R}^{E\times N}$ is the incidence matrix, $W \in \mathbb{R}^{N\times N}$ is the weighted diagonal matrix of the hyper-edge, generally the identity matrix. $\mathbf{X}_h \in \mathbb{R}^{E\times F_1}$ is the input feature matrix, and $\Theta \in \mathbb{R}^{F_1\times F_2}$ is the learnable parameter.

C. Expectation Maximization Algorithm Framework

The Expectation-Maximization (EM) algorithm provides an iterative framework for maximum likelihood estimation in probabilistic models involving latent variables. Given observed data X, latent variables Z, and model parameters θ, the complete-data log-likelihood is defined as:

$$\log p(X, Z \mid \theta) \tag{6}$$

where direct maximization is often intractable due to the presence of unobserved latent variables Z. The EM algorithm iteratively refines parameter estimates through alternating expectation and maximization steps.

In the E-step of iteration t, the expected complete-data log-likelihood is computed using the current parameter estimates $\theta^{(t)}$:

$$Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{Z\sim p(Z|X,\theta^{(t)})}\left[\log p(X, Z \mid \theta)\right] \tag{7}$$

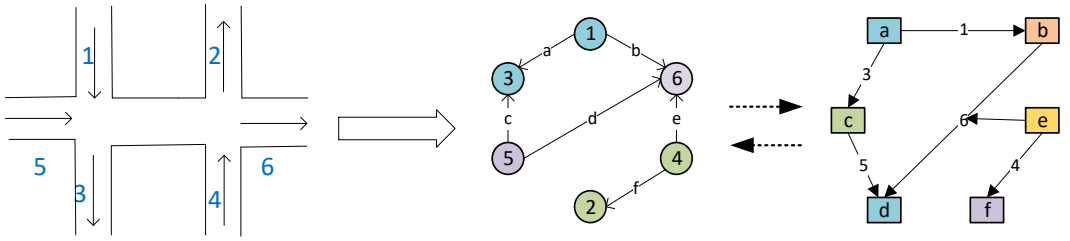
where the expectation is taken with respect to the posterior distribution $p(Z \mid X, \theta^{(t)})$ of the latent variables.

The subsequent M-step updates parameters by maximizing the Q-function:

$$\theta^{(t+1)} = \arg\max_\theta Q(\theta \mid \theta^{(t)}) \tag{8}$$

This alternating optimization scheme ensures a non-decreasing marginal likelihood $p(X \mid \theta)$ at each iteration.

For traffic prediction tasks, the EM framework is implemented to jointly optimize the parameters of the graph learning module θ_g and the prediction network θ_p. By alternately fixing one parameter subset while optimizing the other via Equations (7)–(8), the algorithm enables effective coordination between structural learning and temporal pattern discovery to enhance predictive performance.



(a)Road Network (b)Dual transformation

Fig. 2: The road network and dual transformation.

IV. MODEL

A. Dual transformations for graphs and hypergraphs

The dual graph transformation constitutes a structural conversion where vertices in the primal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ are mapped to edges in the dual hypergraph $\mathcal{G}_h = (\mathcal{V}_h, \mathcal{E}_h, H)$, while primal edges become hypergraph vertices. This operation generates hypergraph structures because nodes with degree > 2 in $\mathcal{G}$ induce hyperedges connecting multiple vertices in $\mathcal{G}_h$. As depicted in Fig. 2(b), the vertex-edge cardinality relationship satisfies:

$$|\mathcal{V}_h| = E, \quad |\mathcal{E}_h| = N \tag{9}$$

The incidence matrix $H \in \mathbb{R}^{E\times N}$ follows the binary convention:

$$[H]_{ij} = \begin{cases} 1 & \text{if } v_i \in e_j \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where $v_i \in \mathcal{V}_h$ and $e_j \in \mathcal{E}_h$. To handle directed traffic networks, we decompose H as:

$$H = H_{\text{src}} + H_{\text{dst}}, \quad H_{\text{src}}, H_{\text{dst}} \in \mathbb{R}^{E\times N} \tag{11}$$

where non-zero elements in H_{src} and H_{dst} encode source/destination nodes respectively.

Hypernode features $\mathcal{X}_h$ in $\mathcal{G}_h$ integrate edge attributes from $\mathcal{G}$ through:

$$\mathcal{X}_h = [(W_1 \odot H_{\text{src}})\mathcal{X};\ (W_2 \odot H_{\text{dst}})\mathcal{X};\ \mathcal{X}_{\text{dis}}] \tag{12}$$

where $\mathcal{X}_h \in \mathbb{R}^{E\times(2F+1)\times T}$, $\odot$ denotes the Hadamard product, $\mathcal{X} \in \mathbb{R}^{N\times F\times T}$ is the node feature matrix, $\mathcal{X}_{dis} \in \mathbb{R}^{E}$ is the road network distance matrix, and $W_1, W_2 \in \mathbb{R}^{E\times N}$ are learnable parameters.The initial components encode dynamic characteristics of directional connections, whereas the final component preserves static relational attributes. Edge-node interdependencies within the traffic network are established through learnable parameter matrices W_1 and W_2.

The inverse conversion operation reconstructs primal node characteristics by projecting hypernode features from the dual domain back to the original traffic graph. This bidirectional mapping preserves structural consistency through:

$$\mathcal{X}' = (W_3 \odot H)^\top \mathcal{X}'_h \tag{13}$$

where $\mathcal{X}' \in \mathbb{R}^{N\times F'\times T}$, $\mathcal{X}'_h \in \mathbb{R}^{E\times F'\times T}$, and $W_3 \in \mathbb{R}^{E\times N}$ is a learnable parameter matrix that captures associations between edges and nodes in the dual hypergraph.

In traffic flow graphs, the number of edges generally exceeds the number of nodes, leading to hypergraphs with a large number of nodes and increasing the complexity of Graph Convolutional Networks (GCNs). Additionally, excessive redundant edges may introduce noise. To address this issue, we select more significant edges from the traffic flow graph to reduce the number of hypergraph nodes. Specifically, a simple Top-k sampling method is applied, where only the top-k edges with the highest node weights are chosen for dual transformation, thereby preserving critical information.

B. Dual hypergraph convolution Module

The underlying mechanism of Graph Convolutional Networks (GCNs) operates through localized information propagation, where nodal representations are iteratively refined by:

$$\mathcal{X}^{(l+1)} = \sigma\left(\hat{A}\mathcal{X}^{(l)}W^{(l)}\right) \tag{14}$$

where $\hat{A}$ denotes the normalized adjacency matrix encoding neighborhood connectivity, $W^{(l)}$ represents trainable parameters at layer l, and $\sigma(\cdot)$ is the activation function. This paradigm systematically assimilates topological information through spectral or spatial aggregation of features from connected nodes within their receptive field.

However, when predicting passenger flow in transit systems, the inter-node correlations exhibit complex patterns where predefined adjacency graphs often fail to provide optimal feature representations for graph learning.

While some scholars have attempted to address the limited expressive capability of geographically adjacent graphs by introducing global semantic adjacency matrices to construct GCN's relational matrices, their theoretical frameworks inadequately utilize the spatiotemporal characteristics inherent in transportation networks. To overcome this limitation, this paper proposes the incorporation of dual hypergraph representations that systematically integrate edge features into the model learning process. This innovative approach ensures deep exploration of edge features while maintaining structural integrity, as demonstrated in Fig. 3. To capture temporal dependencies in traffic data, we employ a Gated Temporal Convolutional Network (Gate-TCN), which utilizes dilated convolution along the temporal dimension for modeling. Given the input feature sequence of traffic flow graph nodes $\mathcal{X} \in \mathbb{R}^{N\times F\times T}$, the Gate-TCN is defined as:

$$\text{TCN}_\phi(\mathcal{X}) = \text{Conv}_g(\mathcal{X}) \in \mathbb{R}^{N\times F\times(T-m(T_0-1))} \tag{15}$$

$$\text{GateTCN}(\mathcal{X}) = \tanh\left(\text{TCN}_{\phi_1}(\mathcal{X})\right) \odot \sigma\left(\text{TCN}_{\phi_2}(\mathcal{X})\right) \tag{16}$$

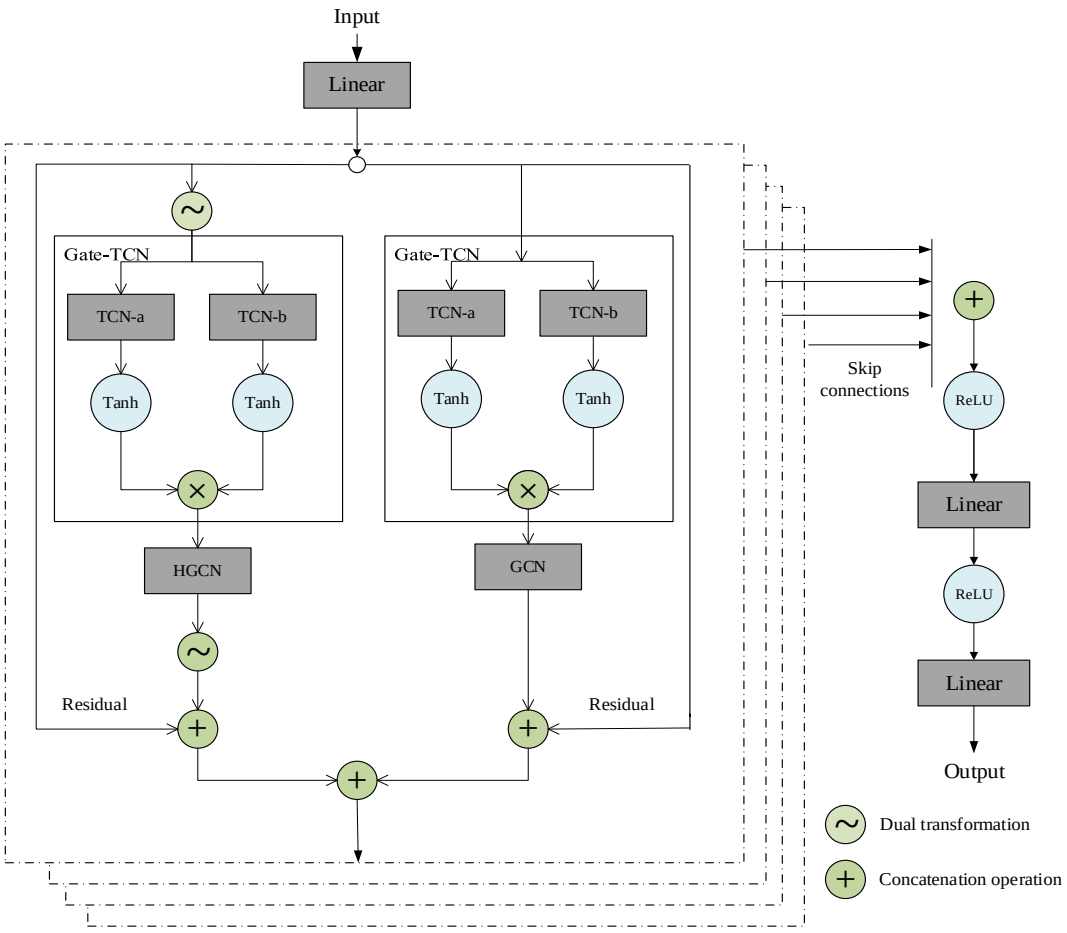


Fig. 3: Dual hypergraph convolution network.

where $\text{GateTCN}(\mathcal{X}) \in \mathbb{R}^{N\times F\times(T-m(T_0-1))}$ denotes the output tensor, $\text{Conv}_g(\cdot)$ represents a dilated convolution operation along the temporal dimension with kernel $g \in \mathbb{R}^{T_0}$ and dilation factor m, $\text{TCN}_{\phi_1}(\cdot)$ and $\text{TCN}_{\phi_2}(\cdot)$ are two independent TCN branches with parameters ϕ_1 and ϕ_2 respectively, and $\tanh(\cdot)$ and $\sigma(\cdot)$ correspond to the hyperbolic tangent and Sigmoid activation functions.

Based on the dual transformation, the hypernode features obtained from Eq.(13) are processed by the Gate-TCN module in Eq.(11), followed by hypergraph convolution through Eq.(5) to capture spatio-temporal features of the hypergraph. Finally, we transform the hypergraph node features via the dual transformation and concatenate them with the node features from the traffic flow graph to accomplish information fusion.

C. Adaptive dual Hypergraph Spatio-Temporal Convolutional Network

1) Adaptive Correlation Graph Learning Module: The spatial dependencies between nodes in dynamic traffic flow are inherently non-bidirectional. As demonstrated in Fig.2(a), upstream traffic fluctuations rapidly influence downstream nodes, while the converse interaction is negligible. To model such directional relationships, we propose an Adaptive Correlation Graph Learning (ACGL) module. A coarse-grained affinity matrix is first derived as:

$$A_1 = \text{ReLU}\left(\alpha \odot (M_1 M_2^\top) - \beta \odot (M_2 M_1^\top) + \text{Diag}(A)\right) \tag{17}$$

where $\alpha, \beta \in \mathbb{R}^N$ denote learnable directional weight vectors, $M_1, M_2 \in \mathbb{R}^{N\times F_0}$ ($F_0 \ll N$) are low-rank projection matrices, and $\text{Diag}(A)$ constructs a diagonal matrix from learnable parameters $A \in \mathbb{R}^N$. The skew-symmetric term $(M_1 M_2^\top - M_2 M_1^\top)$ captures directional spatial interactions, while ReLU activation promotes sparsity by setting the diagonal and half of the off-diagonal entries to zero.

An adaptive fusion mechanism combines real-time and historical dependencies through attention-guided gating:

$$S_t = \text{Softmax}\left(\frac{\mathcal{H}_q(A_1)\cdot \mathcal{H}_k(A_{\text{old}})^\top}{\sqrt{d_k}}\right) \tag{18}$$

where $\mathcal{H}_q(\cdot)$ and $\mathcal{H}_k(\cdot)$ represent independent 1×1 convolutional layers, and d_k is a scaling factor. The final adjacency matrix is computed as:

$$A_{new} = S_t \odot A_1 + (\mathbf{1} - S_t) \odot A_{\text{old}} \tag{19}$$

Here $\odot$ denotes element-wise multiplication and $\mathbf{1}$ is an all-ones matrix. This design enables dynamic weighting between current traffic patterns (A_1) and historical dependencies (A_{old}), prioritizing real-time interactions during peak hours while retaining stable historical correlations.

2) ADHSTGCN Framework : The dual hypergraph convolution demonstrates strong capabilities in capturing edge features and road network characteristics. However, due to the incompleteness of predefined graph structures, it struggles to effectively represent urban road network topologies and track dynamic changes in traffic systems, leading to degraded performance under nonlinear and fluctuating traffic conditions.

To address these challenges, inspired by the Expectation-Maximization (EM) algorithm, we propose an Adaptive Collaborative Graph Learning (ACGL) module. By alternately training the Dual Hypergraph Convolution (DHGC) module and the ACGL module to optimize parameters, we construct a novel model framework named the Adaptive Dual Hypergraph Spatio-Temporal Convolutional Network (ADHGCN).

As shown in Fig. 4, let A^* denote the optimal support matrix that accurately captures spatial dependencies among nodes in the road network structure, and θ^* represent the optimal parameters of the DHGC module. The proposed three-stage optimization process proceeds as follows:

Stage 1: The DHGC module is trained using the current adjacency matrix $\hat{A}_t$ as an approximation of A^*. Through maximum likelihood estimation, this stage yields optimized parameters $\hat{\theta}_t$ for the current iteration:

$$\hat{\theta}_t = \arg\max_{\theta} \mathcal{L}_{\text{DHGC}}(\theta|\hat{A}_t) \tag{20}$$

where $\hat{A}_t$ is subsequently refined via Stage 3 updates. Stage 2: The ACGL module is trained with fixed parameters $\hat{\theta}_t$ (approximating θ^*), generating an enhanced support matrix through sparsity-constrained learning:

$$\tilde{A}_t = \mathcal{F}_{\text{ACGL}}(\hat{A}_t|\hat{\theta}_t) \quad \text{s.t.} \quad \|\tilde{A}_t\|_0 \leq \epsilon \tag{21}$$

where $\mathcal{F}_{\text{ACGL}}(\cdot)$ denotes the transformation with customized loss functions that simultaneously recalibrate dependency intensities and preserve latent information.

Stage 3: The optimized matrix $\tilde{A}_t$ is incorporated into a candidate set $\mathcal{M}$. The support matrix is updated through performance-weighted fusion:

$$\hat{A}_{t+1} = \sum_{A_i \in \mathcal{M}} w_i A_i, \quad w_i \propto \exp(-\eta \mathcal{E}(A_i)) \tag{22}$$

where $\mathcal{E}(\cdot)$ evaluates prediction error and η controls weight concentration. This adaptive aggregation ensures continuous improvement of spatial dependency modeling.

D. Loss Function

1) DHGC Module: As illustrated in Fig. 4, the input layer consists of a linear transformation, whose primary function is to map the input traffic data into a high-dimensional space, thereby enhancing the expressive capability of the network.

The output layer first performs skip-connection operations on all outputs of the DST modules. Subsequently, it fuses these spatiotemporal features and feeds the result into a Leaky ReLU activation function layer and a linear transformation layer, ultimately yielding the prediction result:

$$\hat{X}^{(t+1)}(t+T) = \left(X_{t+1}, X_{t+2}, \ldots, X_{t+T}\right)^T.$$

We employs the Mean Absolute Error (MAE) to construct the loss function. For the ground truth $\hat{X}^{(t+1)}(t+T) = (X_{t+1}, X_{t+2}, \ldots, X_{t+T})^T$, the loss function is defined as follows:

$$\begin{aligned} loss &= \text{MAE}\left(\hat{X}^{(t+1)}(t+T), \hat{X}^{(t+1)}(t+T)\right) \\ &= \frac{\sum_{i=1}^{N}\sum_{j=t+1}^{t+T}|x_j - x'_j|}{N \times T}. \end{aligned} \tag{23}$$

2) AGCL Module: To ensure the sparsity of the generated adjacency matrix, a novel loss function is proposed:

$$\Delta A = \text{ReLU}(A_{\text{new}} - A_{\text{old}}) \tag{24}$$

$$L_G(Y, \hat{Y}) = L_P(Y, \hat{Y}) + \frac{\text{ReLU}\left(\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\Delta A^{(i,j)}}{N^2} - \delta\right)}{\delta} \tag{25}$$

where $\delta \in (0, 1)$ is a hyperparameter controlling the sparsity rate of A_{new}. The smaller δ is, the stronger the sparsity constraint on A_{new}. ΔA calculates the weight difference between the new and old matrices, ensuring that the loss function not only penalizes the number of newly added edges but also suppresses excessive weight growth. This design more precisely suppresses unnecessary high-weight edges, prevents model overfitting, enhances training stability, and the normalization operation effectively constrains the proportion of newly added edges. The spatial dependencies between nodes can be adaptively strengthened or weakened based on the value of L_P, where strong correlations are preserved and weak correlations are gradually eliminated after iterations.

V. EXPERIMENTS

A. Datasets and Baselines

This study is conducted on four public traffic datasets: METR-LA, PEMS-BAY, PEMS03, and PEMS04. METR-LA captures traffic speed data from the Los Angeles metropolitan area, while PEMS-BAY covers the San Francisco Bay Area. The PEMS03 and PEMS04 datasets contain traffic flow data from California Districts 3 and 4, respectively. All datasets include three features—flow, occupancy, and speed—sampled every 5 minutes.

The spatial graphs are constructed based on different criteria: METR-LA and PEMS-BAY use pairwise road segment distances to define edges, whereas the PEMS series datasets rely on road connectivity for graph construction. The adjacency matrices are generated using the thresholded Gaussian kernel to capture spatial correlations. All features are normalized using z-score normalization. Detailed statistics are provided in Table I.

To assess the performance of the proposed model, we conducted experimental trials comparing ADHSTGCN with the following baseline models:

1) Auto Regressive Integrated Moving Average (ARIMA) [4]: A classical model for non-stationary time series, removing trends/seasonality to enable multi-step predictions in complex temporal data.

2) Vector Auto Regression (VAR) [5]: Models multivariate time series by regressing variables on their lagged interdependencies, enabling joint forecasting and causal inference.

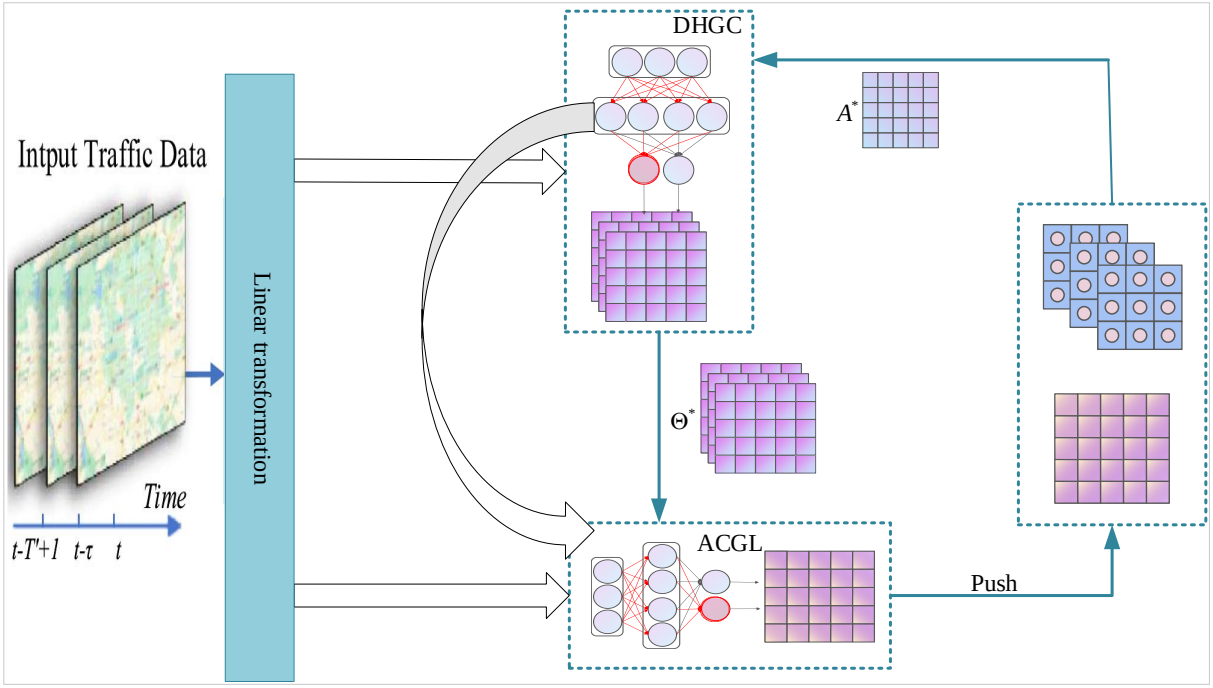


Fig. 4: ADHSTGCN framework, it consists of DHGC module, ACGL module and matrix optimization.The DHGC module stacks multiple layers, each with residual connections, and sums the layer outputs via skip connections.

TABLE I: Basic Statistics of the Datasets Used in the Experiments

Dataset	Location	Time Span	Time Interval	Nodes	Edges	Time Steps	Missingness
PEMS03	CA, USA	9/1/2018 - 11/30/2018	5 min	358	547	26,208	0.67%
PEMS04	CA, USA	1/1/2018 - 2/28/2018	5 min	307	340	16,992	3.18%
PEMS-BAY	CA, USA	1/1/2017 - 5/31/2017	5 min	325	2,369	52,116	0.003%
METR-LA	CA, USA	3/1/2012 - 6/30/2012	5 min	207	1,515	34,272	8.11%

3) Long Short-Term Memory (LSTM) [10]: A variant of RNN, utilizing gates to control information flow and capture long-term dependencies in sequential data.

4) Graph Convolutional Recurrent Network (GCRU) [11]: Combines graph convolutional networks with recurrent layers to jointly capture spatial and temporal dependencies in dynamic graph-structured data.

5) Spatio-Temporal Graph Convolutional Network (STGCN) [29]: Integrates graph convolution with gated temporal units to jointly model spatial dependencies and dynamic traffic patterns in transportation systems.

6) Diffusion Convolutional Recurrent Neural Network (DCRNN) [30]: Combines bidirectional diffusion convolutions with sequence-to-sequence learning to capture spatio-temporal interactions in dynamic networks.

7) Graph WaveNet [32]: Employs dilated causal convolutions and adaptive adjacency matrices to address long-range dependencies and latent spatial relationships.

8) Multi-Range Attentive Bicomponent Graph Convolutional Network (MRA-BGCN) [41]: The model improves traffic forecasting accuracy by integrating multi-range attentive mechanisms with bicomponent graph convolutions to capture complex spatio-temporal dependencies.

9) Spatio-Temporal Self-Supervised Learning (ST-SSL) [35]: The framework enhances traffic pattern representation to reflect spatio-temporal heterogeneity through an assisted self-supervised learning paradigm.

B. Evaluation Metrics

In the empirical validation phase, the predictive capability of our framework is quantified through three principal statistical measures for temporal forecasting: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). These error metrics are formally defined through the following mathematical expressions:

Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{26}$$

Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2} \tag{27}$$

Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Y_i - \hat{Y}_i}{Y_i}\right| \times 100\% \tag{28}$$

where Y_i denotes the actual values, $\hat{Y}_i$ represents the predicted values, and n is the number of samples. Lower values of these metrics indicate higher prediction accuracy. MAE measures absolute error, while RMSE emphasizes overall error trends by penalizing larger deviations. MAPE provides scale-agnostic error comparisons across datasets.

TABLE II: The Traffic Prediction Results of Different Methods on METR-LA and PEMS-BAY

Dataset	Models	15 min			30 min			60 min		
		MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
METR-LA	ARIMA	3.98	8.23	9.62	5.14	10.38	12.76	6.78	13.26	17.67
	VAR	4.33	7.88	10.24	5.33	9.09	12.67	6.23	10.58	15.87
	LSTM	3.44	6.30	9.60	3.77	7.23	10.90	4.37	8.69	13.20
	GCRN	3.56	5.56	8.26	3.54	6.78	10.34	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
	DCRNN	2.77	5.38	7.30	3.15	6.45	8.80	3.60	7.60	10.50
	Graph WaveNet	3.43	6.57	9.73	3.60	6.96	10.35	3.95	7.77	11.65
	MRA-BGCN	2.70	5.14	6.92	3.10	6.96	8.40	3.60	7.77	11.65
	ST-SSL	2.69	5.13	6.90	3.07	6.22	8.37	3.53	7.37	10.01
	ADHSTGCN(ours)	**2.66**	**5.02**	**6.86**	**3.01**	**6.07**	**8.16**	**3.50**	**7.18**	**9.74**
PEMS-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
	LSTM	2.05	4.19	4.80	2.20	4.55	5.20	2.37	4.96	5.57
	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
	DCRNN	1.38	2.95	2.90	1.74	3.97	3.90	2.07	4.74	4.90
	Graph WaveNet	2.54	4.47	5.88	2.60	4.93	6.03	2.71	4.28	4.39
	MRA-BGCN	1.30	2.75	2.75	1.65	3.71	3.70	2.71	3.28	3.39
	ST-SSL	1.30	2.74	2.73	1.63	3.70	3.67	1.95	4.52	4.63
	ADHSTGCN(ours)	**1.29**	**2.68**	**2.70**	**1.61**	**3.68**	**3.65**	**1.90**	**4.37**	**4.46**

TABLE III: Performance Comparison on PEMS03 and PEMS04 Datasets

Models	PEMS03			PEMS04		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
ARIMA	35.41	47.59	33.78	33.73	48.80	24.18
VAR	23.65	38.26	24.51	23.75	36.66	18.09
LSTM	21.33±0.24	35.11±0.50	23.33±4.23	27.14±0.20	41.59±0.21	18.20±0.40
GCRN	19.88±0.04	32.40±0.63	20.78±0.33	26.73±0.66	40.91±0.39	19.20±0.30
STGCN	17.49±0.46	30.12±0.70	17.15±0.35	21.83±0.45	35.55±0.75	14.59±0.21
DCRNN	17.84±0.35	30.31±0.60	17.51±0.40	22.04±0.23	35.72±0.10	15.04±0.10
STSGCN	17.48±0.15	29.21±0.56	16.88±0.20	22.35±0.40	33.65±0.20	14.90±0.05
Graph WaveNet	19.85±0.03	32.94±0.18	19.50±0.15	25.45±0.40	37.70±0.74	17.29±0.24
MRA-BGCN	17.85±0.02	32.84±0.15	18.50±0.15	26.45±0.30	36.70±0.34	16.29±0.24
ST-SSL	17.50±0.15	29.20±0.56	16.78±0.20	21.35±0.40	34.65±0.20	14.90±0.05
ADHSTGCN (ours)	**17.00±0.02**	**27.15±0.43**	**15.54±0.05**	**20.23±0.11**	**31.88±0.05**	**13.70±0.03**

C. Experiment Settings

All experiments are conducted on a Linux server equipped with a 14 vCPU Intel® Xeon® Gold 6330 CPU (2.00 GHz) and an NVIDIA RTX 3090 GPU (24 GB). Following previous studies [32], the datasets are split chronologically into training, validation, and test sets: METR-LA and PEMS-BAY use a 7:1:2 ratio, while the PEMS03 and PEMS04 datasets adopt a 6:2:2 split. The input and output time series lengths are equal, i.e., $T' = T = 12$, indicating that the model predicts the next hour's data using the previous hour's observations. The DHGC module is stacked with $b = 3$ layers. The dilation factor m of Gate-TCN alternates between 1 and 2. All spatial convolutions employ a diffusion coefficient $N = 2$ and incorporate Dropout layers with a probability of $p = 0.3$. The number of intermediate filters is $F = 40$, and top-k sampling uses $k = 4$. The Adam optimizer is applied with a batch size of 64, an initial learning rate of 0.001, and an early stopping strategy to determine termination conditions. For ACGL parameters, the dimensions of matrices M_1 and M_2 are both set to 64, with $\delta = 0.02$. The number of epochs per iteration is adjusted between 5 and 10 based on the convergence speed of the DHGC module.

D. Overall Performance

To systematically evaluate the temporal forecasting capabilities, we developed a multi-granularity evaluation framework with four prediction horizons. The ADHSTGCN model exhibits statistically significant error reduction across all evaluated scenarios ($p < 0.01$, paired t-test). Our experimental design incorporates two complementary data modalities: *instantaneous traffic speed* and *aggregated traffic flow* measurements. To ensure fair benchmarking, we adhere to established protocols from [32] and [35]:

- Speed datasets (METR-LA, PEMS-BAY): Multi-step predictions at 15/30/60-minute intervals
- Flow datasets (PEMS03, PEMS04): Hourly aggregated volume forecasting

The comparative analysis demonstrates that our architecture achieves superior performance in both fine-grained and long-term prediction tasks, particularly in capturing complex spatiotemporal patterns inherent in urban traffic dynamics.

Table II presents the results of different methods for traffic speed prediction, while Table III shows the traffic flow prediction performance across methods. The best results are highlighted in bold, and the second-best results are underlined. Compared to the second-best results, ADHSTGCN achieves average improvements of 1.16%, 1.69%, and 1.87%

in MAE, RMSE, and MAPE metrics, respectively, across three prediction horizons on traffic speed datasets (METR-LA and PEMS-BAY). For traffic flow datasets (PEMS03 and PEMS04), the improvements are more substantial, with average increments of 2.59%, 4.45%, and 3.18% in MAE, RMSE, and MAPE metrics compared to the second-best results.

Tables II and III reveal significant performance disparities among forecasting methodologies. The conventional VAR algorithm delivers suboptimal results across temporal scales, as evidenced by elevated error metrics, which suggest limitations in its reliance on simplistic historical averaging. This fundamental approach proves inadequate for modeling intricate temporal patterns and dynamic trend variations in sequential data.

In contrast, neural architectures employing LSTM and GRU cells achieve enhanced forecasting accuracy with reduced error metrics. However, their sequential processing nature imposes inherent constraints on modeling capacity, particularly manifesting as predictive inaccuracies when handling sophisticated temporal dynamics and extended forecasting horizons.

Graph-based neural architectures such as DCRNN, Graph WaveNet, and STGCN demonstrate marked improvements through sophisticated integration of spatial-temporal correlations. These models surpass conventional approaches by simultaneously processing topological relationships and temporal dependencies. Nevertheless, concerns remain regarding their temporal stability: Graph WaveNet exhibits notable performance degradation (as evidenced by increased error rates) in 60-minute passenger flow predictions, suggesting limitations in capturing long-term temporal dependencies within spatio-temporal graphs. Similarly, STGCN's effectiveness diminishes when processing highly sparse graph structures, potentially due to incomplete adjacency matrix representations.

E. Ablation experiment

To further validate the effectiveness of the components of ADHSTGCN, we conducted ablation experiments by combining the components of the proposed model in different configurations, including the DHGC and ACGL modules, as shown in Fig. 5. The results of one-hour traffic prediction on METR-LA and PEMS03 are presented in Table IV. Specifically, four variants of the model were designed for ADHSTGCN:

1) w/o Dual Hypergraph: This variant uses only a single graph structure (based on the predefined adjacency matrix), removing the dual hypergraph associations and edge feature fusion mechanism.

2) w/o Graph: This variant employs only hypergraphs without leveraging the graph component to verify the effectiveness of the two-channel architecture.

3) w/o DHGC: This variant replaces the DHGC module with a traditional GCN while retaining the ACGL dynamic graph learning mechanism to evaluate the superiority of the dual hypergraph structure.

4) w/o ACGL: This variant utilizes only the DHGC module with fixed predefined graphs, excluding dynamic optimization, to assess the optimization gains provided by the ACGL module.

The ablation study results confirm the effectiveness of key components in the proposed ADHSTGCN model across the METR-LA and PEMS-BAY datasets. The full model (ADHSTGCN) achieves the best performance on both datasets, with MAE values of 3.51 and 17.02, respectively, validating the synergistic integration of dual hypergraphs, dynamic graph learning (ACGL), and hypergraph convolution (DHGC). Notably, removing the dual hypergraph structure (w/o Dual Hypergraph) leads to a significant performance degradation on METR-LA, emphasizing its critical role in capturing edge features and modeling complex urban road networks. However, this degradation is marginal on PEMS-BAY, likely due to the simpler topology of the highway network, where edge features contribute less to spatial dependency modeling. Similarly, disabling the DHGC module (w/o DHGC) causes a 2.6% MAE increase on METR-LA and a 1.2% rise on PEMS-BAY, underscoring the necessity of hypergraph convolution for learning high-order node relationships, particularly in sparse topologies.

The ACGL module demonstrates context-dependent utility: On METR-LA, disabling ACGL (w/o ACGL) results in a 2.0% MAE increase, indicating its capability to refine spatial dependencies in dynamic urban scenarios, while its impact on PEMS-BAY is negligible, suggesting that static graphs suffice for stable highway traffic patterns. Interestingly, completely removing graph structures (w/o graph) degrades METR-LA performance significantly but minimally affects PEMS-BAY, highlighting the heightened importance of spatial modeling in heterogeneous road networks.

These findings reveal that ADHSTGCN's dual hypergraph mechanism and ACGL-driven dynamic adaptation excel in complex, dynamic environments like METR-LA, whereas simpler networks benefit more from baseline graph convolutions. Future work could enhance edge feature engineering for homogeneous networks and explore lightweight ACGL variants to balance efficiency and adaptability across diverse transportation systems.

F. Parameter Sensitivity and Visualization of Experiment Results

1) Parameter Sensitivity: The main hyperparameters of the proposed ADHSTGCN model include the top-k value in the dual transformation, which determines the number of sampled edges, the number of stacked layers b in the DHGC module, and the number of intermediate filters F. To analyze the sensitivity of these hyperparameters, we conduct experiments with different settings on the METR-LA and PEMS03 datasets. The Mean Absolute Error (MAE) is used as the evaluation metric. Detailed results are shown in Table IV.

- Number of stacked layers b: As shown in Table IV, increasing the number of stacked layers b deepens the network, thereby enhancing its representation capacity and improving performance. However, as b continues to increase, the performance gradually stabilizes and no longer exhibits significant improvement. Based on the results, we set $b = 4$ in our model.
- Top-K sampling in dual transformation: K determines the number of hypernodes in the hypergraph. As illustrated in Table IV, increasing K raises the complexity

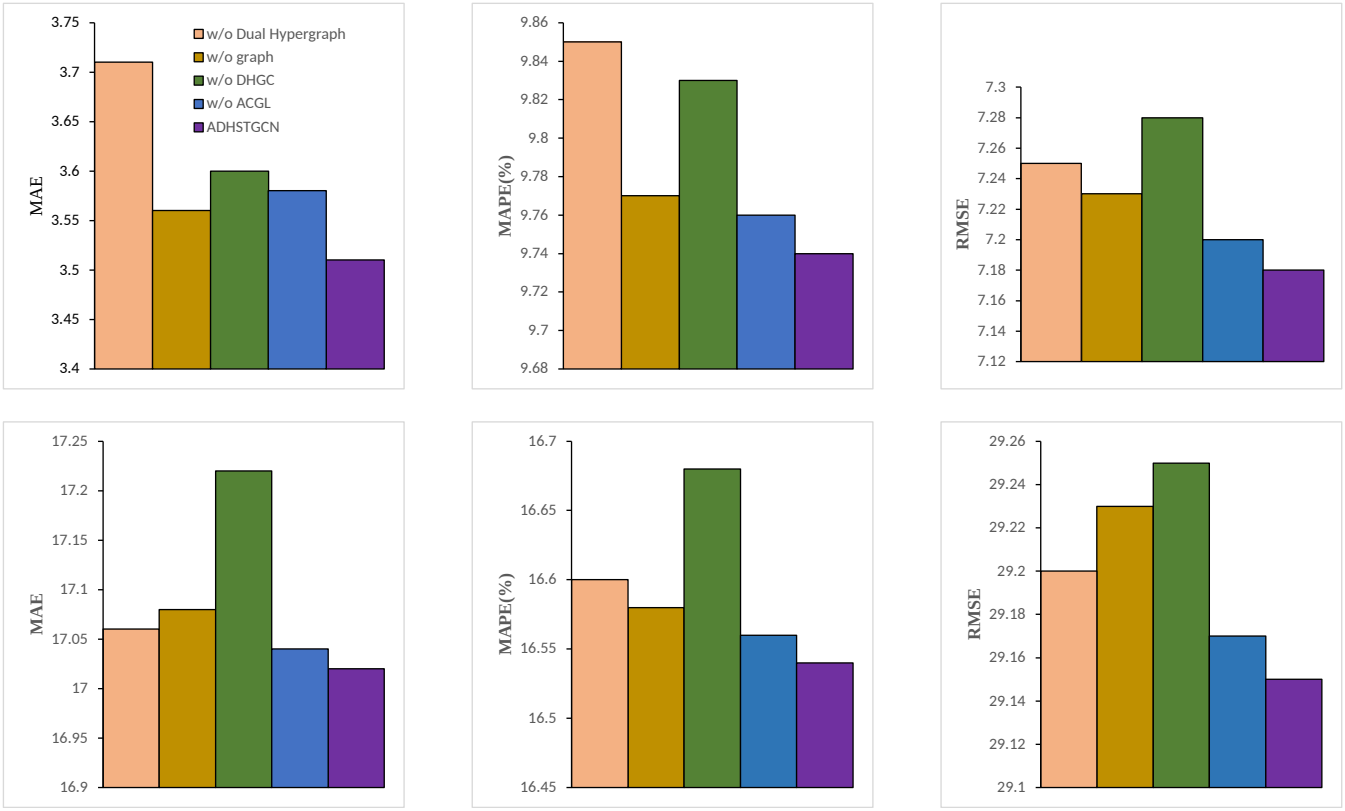


Fig. 5: Ablation Study of ADHSTGCN.

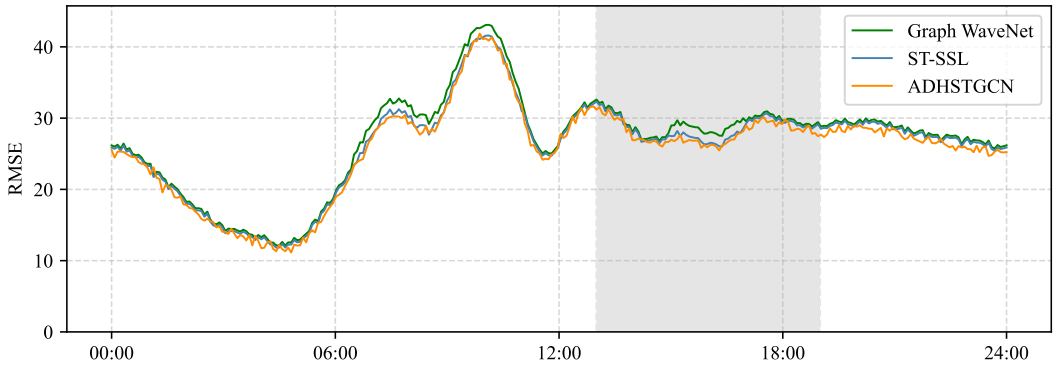
TABLE IV: MAE values for sensitivity analysis of key hyperparameters on model performance.

Parameter value	The number of b layers stacking		The k number of top-k sampling		The number of intermediate filter F	
	METR-LA	PEMS03	METR-LA	PEMS03	METR-LA	PEMS03
1	–	–	3.08	17.00	–	–
2	3.01	15.34	3.03	16.25	–	–
3	2.98	15.40	**2.96**	**15.30**	–	–
4	**2.97**	**15.30**	2.98	15.49	–	–
5	2.96	15.45	3.01	15.24	–	–
6	2.97	15.46	–	–	–	–
30	–	–	–	–	3.00	16.70
35	–	–	–	–	2.99	15.60
40	–	–	–	–	**2.96**	**15.25**
45	–	–	–	–	2.98	15.19
50	–	–	–	–	2.98	15.25

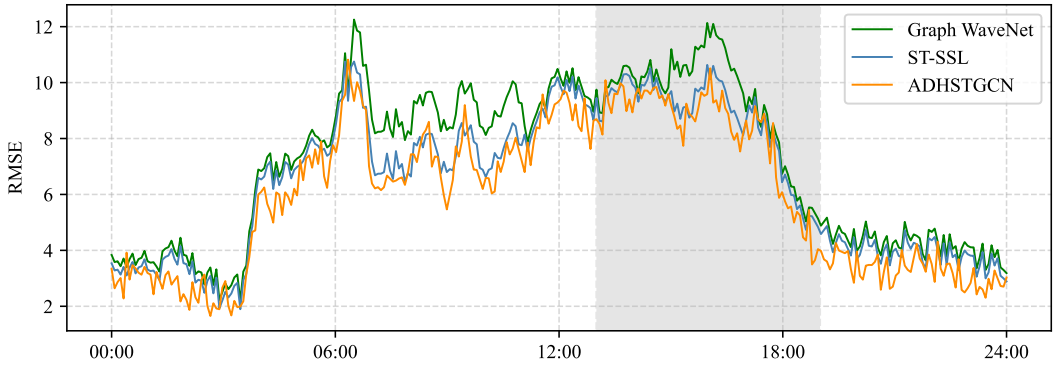
of the dual hypergraph, which enhances its expressive power and improves model performance. Nevertheless, further increasing K introduces excessive noise, leading to degraded performance due to misleading learning signals and increased computational overhead. Therefore, we set $K = 3$ in our experiments.

- Number of intermediate filters F: The parameter F controls the dimensionality of feature channels for nodes and hypernodes in the graph and hypergraph, respectively. As shown in Table IV, the performance initially declines with increasing F, but then gradually stabilizes. However, an excessively large number of filters increases computational cost and may lead to overfitting due to higher model complexity. Consequently, we set $F = 40$ in our implementation.

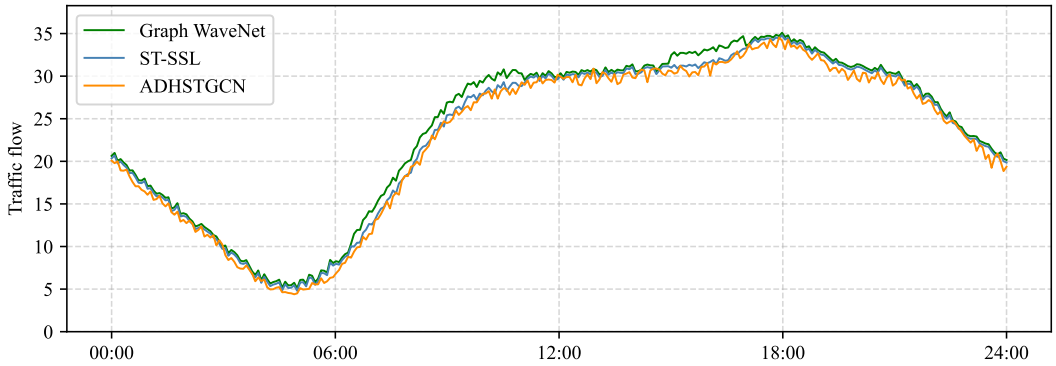
2) Visualization of Experiment Results: To visualize the experimental results more intuitively, we present the daily average traffic prediction outcomes of different methods on the METR-LA and PEMS03 datasets, with comparisons of average RMSE and predicted values against ground truths illustrated in Fig. 6. The results demonstrate the superior performance of our method. In Fig. 6(a), the curve of our ADHSTGCN method lies below those of other approaches, indicating that ADHSTGCN significantly outperforms baseline methods on the PEMS03 dataset. However, the advantage of ADHSTGCN is less pronounced on the METR-LA dataset, as shown in Fig. 6(b). This discrepancy can be attributed to the relatively simpler structure of the METR-LA dataset, which contains fewer nodes and less diverse traffic patterns compared to PEMS03, allowing many methods to achieve competitive results. Notably, during the evening peak hours (highlighted by dark region in Fig. 6(a) and 6(b)), the proposed ADHSTGCN method exhibits significant advantages over all other methods, underscoring its capability to handle complex and dynamic traffic conditions. This observation validates that the synergistic interaction between the



(a) The daily average traffic prediction performance of various methods on PEMS03 evaluated using RMSE.



(b) The daily average traffic prediction performance of various methods on METR-LA evaluated using RMSE.



(c) The daily average traffic prediction curves of different methods on PEMS03.

Fig. 6: The daily average traffic prediction performance of various methods on METR-LA and PEMS03.

DHGC and ACGL modules enables simultaneous modeling of dynamic spatiotemporal features for both nodes and edges in traffic flow graphs. By uncovering deeper attributes of traffic systems, such as congestion propagation dynamics and latent spatial dependencies, the framework enhances robustness in challenging scenarios, solidifying its practical value for real-world traffic management applications.

VI. Conclusion

The proposed ADHSTGCN framework addresses key limitations of current spatiotemporal graph convolutional networks (ST-GCNs) for traffic forecasting, particularly the insufficient exploitation of edge-level features and the unstable training dynamics inherent in end-to-end models. By integrating dual hypergraph associations and an expectation-maximization (EM)-inspired alternating optimization strategy, the model effectively balances static topological constraints with evolving traffic dynamics. Experimental results on METR-LA and PEMS-BAY datasets validate the framework's superiority, demonstrating consistent performance improvements over baseline methods. The dual hypergraph structure proves essential in capturing edge-level interactions, especially in complex urban road networks (METR-LA), where it reduces MAE by 5.7% compared to its ablated variant. This enhancement arises from the model's ability to encode implicit relationships between road segments, such as congestion propagation paths and functional dependencies, which traditional adjacency matrices fail to represent. Simultaneously, the alternating optimization scheme between the DHGC and ACGL modules ensures stable convergence by

alleviating gradient competition typically observed in jointly optimized graph learning and prediction frameworks. This decoupled strategy not only refines spatial dependencies but also uncovers latent node correlations, providing actionable insights for traffic management systems, such as identifying bottleneck intersections or optimizing signal coordination.

The framework's adaptability to diverse network topologies further highlights its practical value. While the ACGL module drives significant gains in dynamic urban environments, its minimal impact on highway networks (PEMS-BAY) reflects context-aware applicability. The integration of edge features within the dual hypergraph, constrained by sparsity regularization, achieves a balance between representational expressiveness and structural interpretability, yielding compact yet semantically rich graph representations. For instance, visual analyses reveal that high-weighted hyperedges frequently align with major arterial roads exhibiting synchronized flow patterns, consistent with established traffic engineering practices.

Future research can extend this paradigm along three primary directions. First, augmenting edge features with multimodal inputs—such as weather data and event logs—may further enhance the dual hypergraph's ability to capture anomalous traffic behaviors. Second, lightweight variants of the ACGL module could be developed for resource-constrained environments, leveraging techniques like graph distillation or dynamic sparsity tuning. Finally, integrating temporal attention mechanisms into the EM optimization loop could enable finer-grained control over graph evolution, particularly for long-term forecasting tasks. The ADHST-GCN framework not only advances the frontier of spatiotemporal graph learning but also provides a modular and extensible blueprint for addressing dynamic urban mobility challenges—ranging from ride-sharing optimization to emergency response routing. By synthesizing theoretical rigor with practical design principles, this work underscores the transformative potential of adaptive graph-based learning in intelligent transportation systems.

REFERENCES

[1] G. E. P. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *J. Amer. Statist. Assoc.*, vol. 65, no. 332, pp. 1509–1526, Apr. 1970.

[2] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transp. Res. C, Emerg. Technol.*, vol. 4, no. 5, pp. 307–318, Oct. 1996.

[3] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using Box–Jenkins techniques," *Transp. Res. Rec.*, vol. 722, pp. 1–9, 1979.

[4] S. Makridakis and M. Hibon, "ARMA models and the box-jenkins methodology," *J. Forecasting*, vol. 16, no. 3, pp. 147–163, May 1997.

[5] D. Qin, "Rise of VAR modelling approach," *J. Econ. Surv.*, vol. 25, no. 1, pp. 156–174, 2011.

[6] J. Liu and W. Guan, "A summary of traffic flow forecasting methods," *J. Highway Transp. Res. Develop.*, vol. 21, no. 3, pp. 82–85, Mar. 2004.

[7] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 10, no. 4, pp. 303–321, Aug. 2002.

[8] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.

[9] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: A general approach," *Transp. Res. B, Methodol.*, vol. 39, no. 2, pp. 141–167, 2005.

[10] Y. Xie, Y. Zhang, and Z. Ye, "Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 22, no. 5, pp. 326–334, 2007.

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, vol. 27, Montreal, QC, Canada, Dec. 2014, pp. 3104–3112.

[12] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Proceedings of the International Conference on Neural Information Processing*, Siem Reap, Cambodia: Springer, 2018, pp. 362–373.

[13] C. Antoniou, H. N. Koutsopoulos, and G. Yannis, "Dynamic data-driven local traffic state estimation and prediction," *Transp. Res. C, Emerg. Technol.*, vol. 34, pp. 89–107, Sep. 2013.

[14] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 3–19, Jun. 2014.

[15] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 50–64, Jun. 2014.

[16] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 62, pp. 21–34, Jan. 2016.

[17] C. Yin, Z. Xiong, H. Chen, J. Wang, D. Cooper, and B. David, "A literature survey on smart cities," *Sci. China Inf. Sci.*, vol. 58, no. 10, pp. 1–18, 2015.

[18] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, 2019.

[19] Z. Li, G. Xiong, Y. Chen, Y. Lv, B. Hu, F. Zhu, and F.-Y. Wang, "A hybrid deep learning approach with GCN and LSTM for traffic flow prediction," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 1929–1933.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, Dec. 2012, pp. 1097–1105.

[21] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, arXiv:1406.1078.

[22] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and deep locally connected networks on graphs," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–14.

[23] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2014.

[24] J. Barros, M. Araujo, and R. J. Rossetti, "Short-term real-time traffic prediction methods: A survey," in *Proc. Int. Conf. Models Technol. Intell. Transp. Syst.*, 2015, pp. 132–139.

[25] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, Nov. 2016, pp. 324–328.

[26] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, p. 1501, Jun. 2017.

[27] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–14.

[28] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 156–165.

[29] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1–7.

[30] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[31] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, arXiv:1803.01271.

[32] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1907–1913.

[33] Z. Diao, G. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial–temporal graph convolutional neural networks for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 890–897.

[34] X. Geng et al., "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 3656–3663.

[35] J. Ji, J. Wang, C. Huang, J. Wu, B. Xu, Z. Wu, J. Zhang, and Y. Zheng, "SpatioTemporal Self-Supervised Learning for Traffic Flow Prediction," *arXiv preprint*, arXiv:2212.04475, 2022.

[36] K. Guo, Y. Hu, Z. Qian, Y. Sun, J. Gao, and B. Yin, "Dynamic graph convolution network for traffic forecasting based on latent network of Laplace matrix estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1–10, Feb. 2020.

[37] J. Li, Y. Liu, and L. Zou, "DynGCN: A dynamic graph convolutional network based on spatial–temporal modeling," in *Proc. Int. Conf. Web Inf. Syst. Eng. Amsterdam, The Netherlands: Springer*, 2020, pp. 83–95.

[38] S. George and A. K. Santra, "Traffic prediction using multifaceted techniques: A survey," *Wireless Pers. Commun.*, vol. 115, no. 2, pp. 1047–1106, Nov. 2020.

[39] J. Ye, J. Zhao, K. Ye, and C. Xu, "How to build a graph-based deep learning architecture in traffic domain: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 3904–3924, May 2020.

[40] K. Guo et al., "Optimized graph convolution recurrent neural network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1138–1149, Feb. 2021.

[41] W. Chen, L. Chen, Y. Xie, W. Cao, and X. Feng, "Multi-range attentive bicomponent graph convolutional network for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, pp. 3529–3536, 2020.

[42] W. Zhang, F. Zhu, Y. Lv, C. Tan, W. Liu, X. Zhang, and F.-Y. Wang, "AdapGL: An adaptive graph learning algorithm for traffic prediction based on spatiotemporal neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 139, p. 103659, Apr. 2022, doi: 10.1016/j.trc.2022.103659.

[43] B. Medina-Salgado, E. Sánchez-DelaCruz, P. Pozos-Parra, and J. E. Sierra, "Urban traffic flow prediction techniques: A review," *Sustain. Comput.: Informat. Syst.*, vol. 35, p. 100739, 2022.

[44] T. Mehari and N. Strodthoff, "Self-supervised representation learning from 12-lead ECG data," *Comput. Biol. Med.*, vol. 141, p. 105202, 2022.

[45] K. Wickstrøm, M. Kampffmeyer, K. Ø. Mikalsen, and R. Jenssen, "Mixing up contrastive learning: Self-supervised representation learning for time series," *Pattern Recognit. Lett.*, vol. 155, pp. 54–61, 2022.

[46] P. Bielak, T. Kajdanowicz, and N. V. Chawla, "Graph barlow twins: A self-supervised representation learning framework for graphs," *Knowl.-Based Syst.*, vol. 256, p. 109631, 2022.

[47] J. Feng, L. Yu, and R. Ma, "AGCN-T: A traffic flow prediction model for spatial-temporal network dynamics," *J. Adv. Transp.*, vol. 2022, p. 1217588, 2022.

[48] S. Zhang, Y. Liu, Y. Xiao, and R. He, "Spatial-temporal upsampling graph convolutional network for daily long-term traffic speed prediction," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8996–9010, 2022.

[49] Y. He, L. Li, X. Zhu, and K. L. Tsui, "Multi-graph convolutional-recurrent neural network (MGC-RNN) for short-term forecasting of transit passenger flow," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18155–18174, 2022.

[50] J. Wu, T. Qiu, H. Tang, and X. Liu, "Network traffic prediction based on a CNN-LSTM with attention mechanism," in *Proc. 7th Int. Conf. Comput. Intell. Appl.*, 2022, pp. 205–209.

[51] J. Hwang, B. Noh, Z. Jin, and H. Yeo, "Asymmetric long-term graph multi-attention network for traffic speed prediction," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2022, pp. 1498–1503.

[52] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 4, 2023.

[53] Z. Liu, Y. Hu, and X. Ding, "Urban road traffic flow prediction with attention-based convolutional bidirectional long short-term memory networks," *Transp. Res. Rec.*, vol. 2677, no. 7, pp. 449–458, 2023.

[54] Y. Oh, M. Jeon, D. Ko, and H. J. Kim, "Randomly shuffled convolution for self-supervised representation learning," *Inf. Sci.*, vol. 623, pp. 206–219, 2023.