ADAPTIVE SPATIO-TEMPORAL GRAPH LEARNING WITH RETRIEVED RECEPTIVE FIELD EXPANSION FOR TRAFFIC FLOW FORECASTING

MS (Research) Thesis

By:

Jatin Sharma



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

November, 2026

Adaptive Spatio-Temporal Graph Learning with Retrieved Receptive Field Expansion for Traffic Flow Forecasting

A THESIS

submitted to the

INDIAN INSTITUTE OF TECHNOLOGY INDORE

in partial fulfillment of the requirements for

the award of the degree

of

MS(Research)

By

Jatin Sharma



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

May 2026

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Adaptive Spatio-Temporal Graph Learning with Retrieved Receptive Field Expansion for Traffic Flow Forecasting** in the partial fulfillment of the requirements for the award of the degree of **MS(Research)** and submitted in the **Department of Computer Science and Engineering, Indian Institute of Technology Indore,** is an authentic record of my own work carried out during the time period from August 2024 to May 2026.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Jatin 22/05/2026

Signature with Date
(Jatin Sharma)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Nagendra Kumar 19/11/2026

Signature of the Supervisor of
M.S. (Research) Thesis

(Dr. Nagendra Kumar)

Jatin Sharma has successfully delivered the M.S. (Research) Oral Examination held on 19th November 2026.

ACKNOWLEDGEMENTS

I am deeply grateful for the opportunity to extend my sincere appreciation to several individuals whose contributions have made this journey both enriching and manageable. First and foremost, I express my gratitude to my supervisor, Dr. Nagendra Kumar, whose unwavering guidance and inspiration have been pivotal throughout this endeavour. Without his steady direction and invaluable insights, this research endeavour would not have reached fruition. His continuous support and encouragement have served as a beacon, guiding me through the intricacies of this work.

I am thankful to my research committee member for taking out some valuable time to evaluate my progress throughout the course. Their good comments and suggestions helped me to improve my work at various stages. I am also grateful to Dr. Ranveer Singh, HOD of Computer Science and Engineering, for his help and support.

My sincere acknowledgement and respect to Prof. Suhas S. Joshi , Director, Indian Institute of Technology Indore for providing me the opportunity to explore my research capabilities at Indian Institute of Technology Indore.

My deepest gratitude goes to my family and friends for their unwavering love and support throughout the process. Their encouragement and understanding during challenging times were invaluable

Lastly, I extend my thanks to all those who have directly or indirectly contributed, assisted, and supported me on this path of academic pursuit.

Jatin Sharma

Abstract

Traffic Flow Prediction is challenging due to its complex heterogeneous structure and simultaneously evolving spatial and temporal dependencies. Current traffic forecasting methods are constrained by fixed input lengths, making them vulnerable to spatio-temporal illusions, where similar short-term observations result in dissimilar long-term outcomes, leading to severe ambiguity caused by limited contextual awareness. Moreover, most existing traffic forecasting frameworks rely on a single-view spatial representation which fails to capture the complexity of urban networks, where multi-hop dependencies are influenced by both the physical topology and the inherent randomness of traffic flow. To address these limitations, we propose Adaptive Spatio-Temporal Graph Learning with Retrieved Receptive Field Expansion (STGLRNet) which utilizes dynamic temporal retrieval and multi-view spatial modeling. First, we introduce a Memory-Enhanced Feature Extraction Unit (MFEU) that dynamically retrieves extended historical context, thereby overcoming fixed receptive field constraints. Second, a Hierarchical Pattern Recognition Network (HPRN) is designed to extract deep representations by jointly modeling multi-view spatio-temporal patterns through adaptive, cyclic, and critical node-aware adjacency matrices. Third, a Dual-Stream Attention Network (DSAN) with learnable embeddings and an attention mechanism is employed to learn the inherent structure of the spatio-temporal system. Extensive experiments on six widely used benchmark datasets, PEMS03, PEMS04, PEMS07, PEMS08, METR-LA, and PEMS-BAY, were conducted to evaluate our proposed methodology.

List of Publications

Journal Papers Published

[1] *Graph-Agnostic Temporal Pretraining for Multi-City Traffic Forecasting*
Published in *Physica A: Statistical Mechanics and its Applications*. doi:10.1016/j.physa.2026.132143.

Journal Papers Under Review

[1] *Adaptive Spatio-Temporal Graph Learning with Retrieved Receptive Field Expansion for Traffic Flow Forecasting*
Submitted to *Applied Soft Computing* on 11 October 2026.

[2] *LTFNet: Latent Traffic Field Networks for Spatio-Temporal Traffic Forecasting*
Submitted to *The Journal of Supercomputing* on 31 March 2026. Currently under editorial review.

Contents

List of Publications iii

List of Figures vii

List of Tables ix

List of Abbreviations and Acronyms xi

1 Introduction 1

1.1 Research Problem 3

2 Literature Review 5

2.1 Statistical and Machine Learning Methods 5

2.2 GNN based Methods 6

2.3 Transformer-based Methods 6

3 Problem Statement and Methodology 9

3.1 Problem Definition 9

3.2 Methodology 10

3.2.1 Memory-Enhanced Feature Extraction Unit 10

3.2.2 Hierarchical Pattern Recognition Network 13

3.2.3 Dual-Stream Attention Network 17

3.2.4 Integrated Prediction Network 18

4 Experimental Evaluations and Result 19

4.1 Datasets and Evaluation Metrics 19

4.2 Exploratory Data Analysis (EDA) . . . 20
4.3 Compared Methods . . . 22
4.3.1 Classical & Statistical Methods . . . 22
4.3.2 Classic Spatio-Temporal GNN Methods . . . 22
4.3.3 Adaptive Graph-Learning Methods . . . 22
4.3.4 Transformer-Enhanced Spatio-Temporal Methods . . . 23
4.4 Experimental Settings . . . 23
4.5 Experimental Results . . . 25
4.5.1 Performance Comparison on Traffic Flow Datasets . . . 25
4.5.2 Comparison of Performance on Traffic Speed Datasets . . . 27
4.5.3 Performance Gain Analysis . . . 29
4.5.4 Qualitative Analysis and Pattern Visualization . . . 31
4.5.5 Hyperparameter Sensitivity Analysis . . . 32

5 Discussion **35**
5.1 Temporal Database Scalability and Index Management . . . 35
5.2 Scalability and Deployment Considerations . . . 36
5.3 Limitations and Future Work . . . 37
5.4 Beyond Retrieval-Based Forecasting: Continuous Latent Dynamics . . . 38
5.5 Towards Transferable and Topology-Agnostic Forecasting . . . 39

6 Conclusion **41**

Bibliography **48**

List of Figures

1.1 The figure showing how traffic sensors are linked, some connections stay the same, others change with the flow. Each numbered cylinder is a sensor. The red part of its graph shows what happened before; the green part is what we're trying to predict. Gray arrows mean "these sensors are always connected", like roads that don't move. Pink arrows mean "they influence each other right now, but maybe not after some time". 2

3.1 Proposed architecture comprising four blocks: (A) Memory-Enhanced Feature Extraction Unit, (B) Hierarchical Pattern Recognition Network, (C) Dual-Stream Attention Network, and (D) Integrated Prediction Network. 11

3.2 Capturing transitive spatio-temporal dependencies via recursive temporal retrieval. 11

4.1 Daily traffic flow trends on different sensors of the PEMS08 dataset. . . 20

4.2 Weekly traffic flow trend on PEMS08 dataset. 21

4.3 Spatio-Temporal illusions in Traffic flow of PEMS08 dataset. 21

4.4 Predictions V/s Ground Truth over our proposed method with five baselines for 300 steps on Sensor Number 150 of the PEMS08 Dataset. . . . 28

4.5 Horizon-wise evaluation of MAE, RMSE, and MAPE on the PEMS08 dataset. 29

4.6 Predicted vs. Ground Truth comparison for 3000-step prediction on the PEMS08 dataset. 31

4.7 Dynamically Changing Spatio-Temporal Patterns. 32

4.8 Sensitivity of performance metrics (MAE and RMSE) to R (i.e., receptive field length) on the PEMS08 dataset. 33
4.9 Sensitivity of performance metrics (MAE and RMSE) to k (i.e., number of critical nodes) on the PEMS08 dataset. 34

List of Tables

4.1 Statistical summary of the benchmark datasets. 20
4.2 Performance comparison on PEMS03 and PEMS04 using MAE, RMSE, and MAPE across baselines. 25
4.3 Performance comparison on PEMS07 and PEMS08 using MAE, RMSE, and MAPE across baselines. 26
4.4 Performance comparison of methods on METR-LA and PEMS-BAY datasets. 29
4.5 Hierarchical ablation study of STGLRNet. 30

List of Abbreviations and Acronyms

DTW Dynamic Time Warping

GCN Graph Convolutional Network

GNN Graph Neural Networks

GRU Gated Recurrent Unit

GATP Graph-Agnostic Temporal Pretraining for Multi-City Traffic Forecasting

ITS Intelligent Transportation Systems

LTFNet Latent Traffic Field Networks for Spatio-Temporal Traffic Forecasting

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

MLP Multilayer Perceptron

RMSE Root Mean Square Error

STGLRNet Adaptive Spatio-Temporal Graph Learning with Retrieved Receptive Field Expansion for Traffic Flow Forecasting

TTT Test Time Training

Chapter 1

Introduction

Rapid urbanization is placing significant strain on urban infrastructure, as exponentially growing traffic volumes lead to severe congestion, transportation delays, and reduced system performance. Consequently, it becomes crucial to utilise existing infrastructure to its full potential, providing sustainable solutions to urban traffic problems. This necessity has spurred the development of Intelligent Transportation Systems (ITS) [1]. ITS rely mainly on three core factors: accurate traffic flow prediction, on-the-spot data collection via sensors and connected devices, and the facilities of efficient communication and control. The resulting combination gives precise decision-making, which facilitates the implementation of measures to ease the traffic congestion problem, make efficient use of the road network, and manage traffic signals in an intelligent way [1]. As represented in Fig. 1.1, Traffic Flow forecasting is a classic example of a Spatio-Temporal Prediction problem, which remained challenging over the years due to its complex heterogeneous structure, i.e., Spatial and temporal dependencies simultaneously changing over time, making it hard to predict. We observe spatial dependencies ranging from local to global levels. Similarly, in the temporal dimension, both short- term and long-term dependencies exist, making it difficult to interpret exact predictable patterns. Initial works used averages based methods like Historical Average (HA) [2] and Auto-Regressive Integrated Moving Average (ARIMA) [3] which were good at giving statistical estimates of future traffic but were not accurate at all, Researchers also tried Traditional machine learning

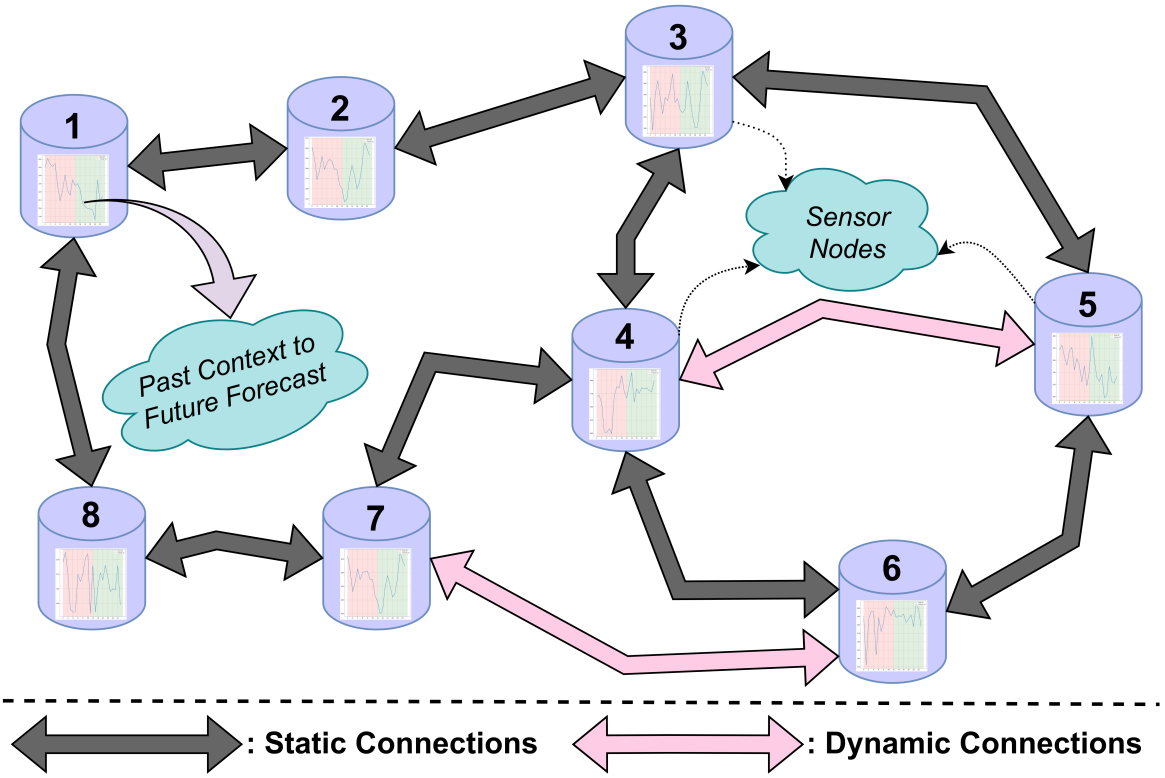


Figure 1.1: The figure showing how traffic sensors are linked, some connections stay the same, others change with the flow. Each numbered cylinder is a sensor. The red part of its graph shows what happened before; the green part is what we're trying to predict. Gray arrows mean "these sensors are always connected", like roads that don't move. Pink arrows mean "they influence each other right now, but maybe not after some time".

based methods like Support Vector Regression (SVR) and Vector Auto-Regression (VAR) [4] to capture temporal trend and learning non linear relationships. Recent Works, including Spatio-Temporal Graph Convolution (STGCN) [5], Graph Wavenet (GWNet) [6], and Diffusion Convolution Recurrent Neural Network (DCRNN) [7], etc, tried to model both spatial and temporal dependencies, but due to the limited input information, these methods fall into spatio-temporal illusions, i.e., for a similar-looking past input sequences we may have dissimilar future sequences.

Most existing traffic prediction methods fail to effectively capture the joint static and dynamic dependencies within traffic sequences. Moreover, most of the recent works suffer from limited input information, complex spatio-temporal dependencies, and challenges in handling multi-view correlations. These limitations motivate the need

for methods that can comprehensively capture both spatial and temporal patterns while efficiently integrating multiple views of traffic data.
Moreover, Traffic data has an inherent periodic nature. (i.,e daily periodicity and weekly periodicity). But it can be shown that daily periodicity is the dominant one.

1.1 Research Problem

Traffic flow forecasting remains challenging because traffic dynamics exhibit complex temporal dependencies and spatial correlations that evolve. Despite advances in spatio-temporal graph learning, three key limitations remain.

First, to overcome the limitations of fixed receptive fields, which constrain the model's ability to capture long-range dependencies and may introduce spatio-temporal distortions, we introduce a causal, time-indexed vector memory mechanism. With our new method, instead of using a fixed look-back window, the model regards the present input sequence as a key to fetch the most appropriate historical data of the required length from the database. This approach can be seen as a way to enrich the model by adding some prior information.
Secondly, we introduce a Hierarchical Pattern Recognition Network to recognise the static and dynamic traffic patterns simultaneously, as the performance of the prediction network relies on the diverse traffic patterns, which are combinations of long-term and short-term dependencies. The network is designed to extract and process multi-view patterns by jointly modelling static and dynamic correlations over time.
Thirdly, we learned periodic patterns (i.e., daily and weekly) using embedding layers and learned them effectively using a dual-stream attention mechanism, which works on spatial and temporal dimensions in parallel, such that our learned embeddings have subtle representations of both domains.
The proposed framework is evaluated on six benchmark datasets—PEMS03, PEMS04, PEMS07, PEMS08, METR-LA, and PEMS-BAY—using standard traffic forecasting metrics. The detailed mathematical formulation and architectural design of these

components are presented in Chapter 3, while their empirical evaluation is provided in Chapter 4.

In this thesis, we address these drawbacks by introducing a new deep neural network framework centred on three fundamental ideas to comprehend traffic flow more flexibly and completely. We mainly make the following contributions:

[1] We propose a dynamic receptive field that is based on a retrieval approach, which is the solution to the problem of fixed-length input sequences in conventional methods.

[2] We propose and design a Hierarchical Pattern Recognition Network that not only unveils but also aggregates the deep traffic patterns.

[3] We present an embedding-based dual stream attention mechanism that adeptly learns the periodic patterns.

[4] We conduct comprehensive assessments on six benchmark datasets, showing the potential of our method.

Chapter 2

Literature Review

Traffic flow prediction is one of the main cases for time-series problems, which have been treated in a large measure. The way to treat these problems evolved from the use of traditional statistics methods to complex deep learning architectures targeting to capture spatio-temporal dependencies of the data.

2.1 Statistical and Machine Learning Methods

Statistical methods were the basis of early traffic forecasting methods. Several types of methods, like Historical Average (HA) [3], Auto-Regressive Integrated Moving Average (ARIMA) [3], and Vector Auto-Regression (VAR) [8], were in the leading positions during those times. Even though such methods are capable of scoping the temporal aspect of the data, they are usually limited in their capacity to express spatial relationships within road networks, which are complex and non-linear in nature. Traditional machine learning methods like Support Vector Regression (SVR) [8] have also been tried and have yielded better performance; however, there still exist problems in completely capturing spatial and temporal dynamics in unison. In addition to this, these methods are still used as the base for comparison in current research.

2.2 GNN based Methods

The turnaround in deep learning, as well as the technology of Graph Neural Networks (GNNs), signified a major leap for traffic prediction. By introducing road network graphs, GNNs can uncover spatial dependencies. To capture spatial and temporal features, many researchers combine GNNs with the aid of deep recurrent or convolutional networks.

The DCRNN [7] first steps were based on diffusion convolution on the directed graph combined with the recurrent neural network. STGCN [5] proposed a quite different integration of the graph convolution with 1D temporal convolutions. After that, many variants have come up. The idea of Graph WaveNet (GWNet) [6] was to learn the hidden spatial relations via an adaptive adjacency matrix that is adaptive. A similar aim is addressed by MTGNN [9] and GTS [10], i.e., these methods learn the graph structure from the data only. Along with AGCRN [2], where the adaptive modules are used both for the graph convolution and the recurrent units, and DGCRN [11], which is focused on dynamically changing the graph structures, there are other GNN-based methods. STEMGNN [12] is radically different in that it solves the problem in the frequency domain.

2.3 Transformer-based Methods

The development of the attention mechanism, the key idea of the Transformer architecture, has led to its successful applications in spatio-temporal forecasting, besides all the other fields. The long-range dependency issue is one of the strong points of these methods. STID [13] suggested a baseline that was easy but powerful, using both spatial and temporal identity information within the Transformer framework. STEP [14] improved the results by adding the pre-training strategy. There have been several attempts to adapt the main attention mechanism to the special features of traffic data. For instance, STWave [15] combines attention with wavelet to make analysis, and STAEformer [16] and STGAFormer [17] introduce specifically designed spatio-

temporal attention units. Furthermore, there is also ST-Norm [18], which focuses on normalisation techniques, and STDMAE [19], a masked autoencoder pre-training scheme.

Chapter 3

Problem Statement and Methodology

This chapter presents the formal problem definition of traffic flow forecasting and describes the proposed methodology adopted in this work. It introduces the mathematical formulation of the forecasting task, the traffic network representation, and the overall architecture of the proposed framework, including its major components and underlying design principles.

3.1 Problem Definition

Given a road network represented as a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{A})$ where $\mathbf{V}$ is the set of N nodes, $\mathbf{E}$ represents the edges connecting nodes, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix. Let $\mathbf{X}^{(t)} \in \mathbb{R}^{N \times d}$ denote the traffic measurements at time step t across all nodes, where d is the number of features (e.g., speed, flow). The traffic prediction problem is formulated as learning a mapping function, as shown in Eq. 3.1:

$$f : (\mathbf{G}, \mathbf{X}_{t-L+1:t}) \rightarrow \hat{\mathbf{X}}_{t+1:t+H} \tag{3.1}$$

Given historical observations $\mathbf{X}_{t-L+1:t} \in \mathbb{R}^{N \times L \times d}$ of length L, the goal is to predict future traffic states $\hat{\mathbf{X}}_{t+1:t+H} \in \mathbb{R}^{N \times H \times d}$ for the next H time steps.

3.2 Methodology

This section delves into the technical specifics of our proposed Adaptive Spatio-Temporal Graph Learning with Retrieved Receptive Field Expansion for Traffic Flow Forecasting (STGLRNet) framework, as delineated in Fig. 3.1. The framework is divided into four major blocks addressing the critical challenges outlined in section 1. The first block is the Memory-Enhanced Feature Extraction Unit. It retrieves long-context sequences and uses them to help short input sequences benefit from the wider context. This is done through cross-attention between the short input and the retrieved long-context sequences. Next, we have the Hierarchical Pattern Recognition Network, which is based on a Graph Neural Networks (GNN) and captures multi-view spatio-temporal correlations by using multiple adjacency matrices for deeper pattern extraction. Furthermore, the Dual-Stream Attention Network is employed to model temporal periodicities and inter-series non-graphical dependencies. Finally, we have the Integrated Prediction Network, which combines the hidden feature representations and decodes them into the final output using a regression layer. The following subsections explain each part of the design, showing what it does and how it helps the whole system work.

3.2.1 Memory-Enhanced Feature Extraction Unit

The predictive performance of a traffic forecasting model depends on the historical context available to its feature extraction mechanism. Conventional methods typically operate on a fixed-length input window, which may not contain sufficient information to distinguish traffic sequences that appear similar in the short term but evolve differently in the future. Recent developments in deep learning, such as Retrieval-Augmented Generation (RAG)[20], have shown potential to solve the above issue. Inspired by this, we introduce a novel Memory-Enhanced Feature Extraction Unit (MFEU), depicted as Block A in Fig. 3.1. This module has two Components: 1. Receptive Field Retriever, a causal vector database for continuous retrieval and integration of sequences during training, as well as inference. 2. Memory Guided At-

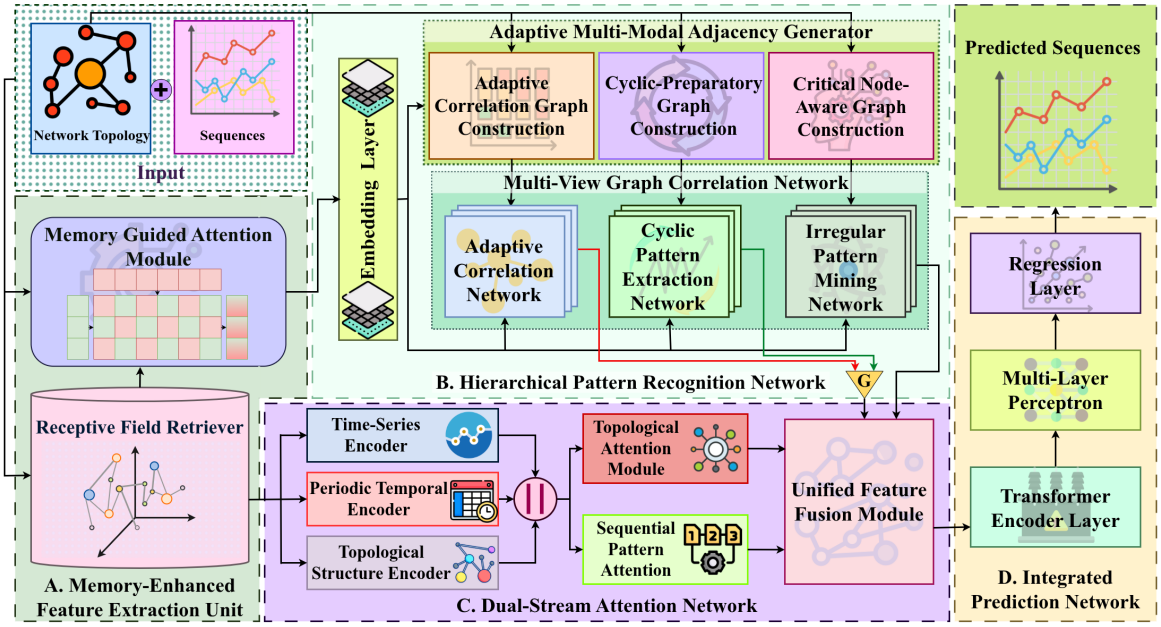


Figure 3.1: Proposed architecture comprising four blocks: (A) Memory-Enhanced Feature Extraction Unit, (B) Hierarchical Pattern Recognition Network, (C) Dual-Stream Attention Network, and (D) Integrated Prediction Network.

tention Module for learning the useful context of retrieved sequences with given short input sequences using cross attention.

3.2.1.1 Receptive Field Retriever

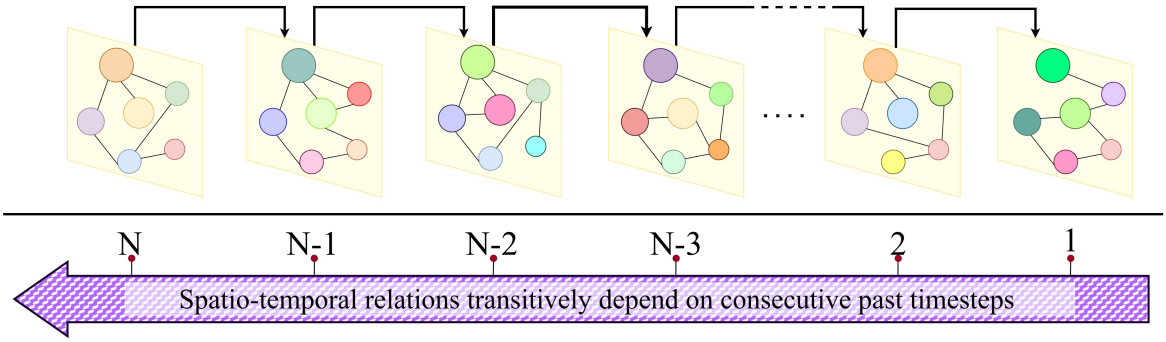


Figure 3.2: Capturing transitive spatio-temporal dependencies via recursive temporal retrieval.

Time series data are inherently autoregressive, every temporally succeeding observation is causally dependent on its history. We define a temporal embedding database

that retains the causal ordering of a data stream using indexing by temporal appearance and keeping dependencies along sequential chains. As illustrated in Fig. 3.2, traffic flow patterns manifest distinct unidirectional dependencies across time. The current traffic state $\mathbf{X}^{(t)} \in \mathbb{R}^{N \times d}$ is functionally dependent upon its preceding states $\{\mathbf{X}^{(t-1)}, \ldots, \mathbf{X}^{(t-k)}\}$, subsequently forming a chain of temporal dependencies in a continuous process. A retrieval-augmented mechanism is designed to focus on the architectural building blocks of transitive dependency structures intrinsic within sequential data.

The temporal database constructed here acts like a dynamic repository of memory, allowing for the retrieval of an extended historical context given a short input sequence. For each input sequence $\mathbf{X}_{\text{in}} \in \mathbb{R}^{N \times L \times d}$ of length L, we embed it into a latent representation space via a learned encoder $f_\theta : \mathbb{R}^{N \times L \times d} \to \mathbb{R}^{d'}$ producing $\mathbf{z}_t$. The encoded representation is indexed in the temporal database with its timestamp t, yielding a vectorised entry retaining both semantic content and temporal location. During retrieval, we utilise FAISS (Facebook AI Similarity Search) [21] for efficient approximate nearest-neighbour searches within the embedding space. We first take a query sequence $\mathbf{X}_{\text{in}}$ of length L and vectorise it into a flat representation $\mathbf{z}_t$ and query the indexed temporal database to get the most similar historical pattern at time stamp t^*.Formally, we retrieve the corresponding timestamp t* with the nearestneighbor criterion as defined in Eq. 3.2:

$$t^* = \arg\min_{t_i < t} \|\mathbf{z}_t - \mathbf{z}_{t_i}\|_2 \tag{3.2}$$

where the constraint $t_i < t$ enforces causal consistency by ensuring that only past observations are considered. The number of retrieved historical time steps is denoted by T_{ret}. Upon identifying the matching timestamp t^*, we extract an extended context window $\mathbf{C}_t = [\mathbf{X}^{(t^*-T_{\text{ret}})}, \ldots, \mathbf{X}^{(t^*+L-1)}] \in \mathbb{R}^{N \times (T_{\text{ret}}+L) \times d}$ comprising T_{ret} preceding steps and the matched L-length sequence. To maintain coherence in contextual access, the retrieval process is implemented identically across both training and inference stages, thereby preserving consistency in how historical information is utilized by the

model. The FAISS index is built offline by first embedding all historical sequences $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}$ from the temporal database into vector representations and subsequently indexing them. By precomputing the index, the system achieves sub-linear time complexity for similarity searches across the full history. At inference time, it further supports continuous learning by incrementally adding newly observed sequences to the index via FAISS's dynamic insertion mechanism, thereby allowing the temporal memory to evolve in real time.

3.2.1.2 Memory-Guided Attention Module

While the retrieval mechanism provides access to extended historical context of length $T_{\text{ret}} + L$, not all retrieved information is equally relevant to the current prediction task. To distill the most salient temporal patterns and mitigate computational overhead, we employ a cross-attention mechanism where the input query sequence $\mathbf{X}_{\text{in}}$ attends to the retrieved context $\mathbf{C}_t$. This selective attention mechanism compresses the extended context into the length representation of size L, preserving only the most contextually relevant historical patterns while maintaining constant computational complexity for downstream processing. By dynamically filtering the retrieved memory, we achieve the benefits of long-range temporal modeling without incurring quadratic scaling costs.

3.2.2 Hierarchical Pattern Recognition Network

Traffic networks have complex spatio-temporal dependencies. These manifest as dynamic correlations that change over time, periodic patterns that repeat, and irregular fluctuations caused by critical events. Traditional methods employ static graph topologies, which are unable to capture these dynamic interactions. To address this, we introduce the Hierarchical Pattern Recognition Network (HPRN), shown as Block B in Fig. 3.1. HPRN methods adaptive, cyclical, and critical traffic patterns using a multi-view graph representation.

The core principle of HPRN is to learn spatial dependencies into three complemen-

tary views, each captured by a specialised adjacency matrix that encodes a distinct correlation structure. We first present the Adaptive Multi-Modal Adjacency Generator, which constructs these three adjacency representations. This is followed by the Multi-Adjacency Graph Correlation Network, which performs unified spatial reasoning across all views.

3.2.2.1 Adaptive Multi-Modal Adjacency Generator

Given input sequences $\mathbf{X} \in \mathbb{R}^{N \times T \times d}$ and road network topology $\mathbf{G}$, where N denotes the number of sensors, T the temporal length, and d the feature dimension, we construct three complementary adjacency matrices that capture distinct correlation modalities between nodes: (a) Adaptive Correlation Graph Construction (ACG), which learns data-driven dynamic dependencies; (b) Cyclic-Preparatory Graph Construction (CPG), which encodes temporal pattern similarities across daily cycles; and (c) Critical Node-Aware Graph Construction (CNG), which emphasizes interactions among highly fluctuating nodes. The construction methodology for each adjacency is detailed below.

3.2.2.1.1 Adaptive Correlation Graph Construction Static adjacency matrices assume fixed spatial relationships, yet traffic correlations evolve dynamically with changing conditions such as congestion propagation and incident cascades.
The ACG mechanism addresses this by learning time-dependent inter-sensor dependencies directly from data. For each node i, we obtain temporal embeddings via a Gated Recurrent Unit (GRU) that encodes sequential patterns as shown in Eq. 3.3:

$$\mathbf{h}_t^{(i)} = \mathrm{GRU}(\mathbf{x}_t^{(i)}, \mathbf{h}_{t-1}^{(i)}) \tag{3.3}$$

where $\mathbf{x}_t^{(i)}$ represents the input features of sensor i at time t. These embeddings are projected into a relational space through learned transformations $\mathbf{W}_K$ and $\mathbf{W}_Q$, producing key and query representations: $\mathbf{K}_i = \mathbf{h}^{(i)}\mathbf{W}_K$ and $\mathbf{Q}_j = \mathbf{h}^{(j)}\mathbf{W}_Q$. Pairwise

attention scores quantify the influence of node i on node j, as shown in Eq. 3.4:

$$E_{ij} = \text{LeakyReLU}(\boldsymbol{\alpha}^\top [\mathbf{K}_i \| \mathbf{Q}_j]) \tag{3.4}$$

where $\boldsymbol{\alpha}$ is a learnable attention vector and $\|$ denotes concatenation. To ensure bidirectional consistency, we symmetrize the attention matrix: $\mathbf{A}_{\text{dyn}} = \frac{1}{2}(E + E^\top)$, yielding an adaptive adjacency that evolves with traffic dynamics.

3.2.2.1.2 Cyclic-Preparatory Graph Construction Traffic patterns exhibit strong periodicity, with similar behaviors recurring across daily, weekly, and seasonal cycles. Nodes that share similar temporal signatures often experience correlated conditions, even without direct spatial proximity. The CPG captures these cyclical similarities by measuring temporal pattern alignment between nodes. For each time step t, we extract the historical time series $\mathbf{s}_i^{(t)} \in \mathbb{R}^H$ of length H for node i, and compute pairwise temporal distances using Dynamic Time Warping (DTW) as shown in Eq. 3.5, which handles phase shifts and temporal misalignments:

$$d_{ij}^{(t)} = \text{DTW}(\mathbf{s}_i^{(t)}, \mathbf{s}_j^{(t)}) \tag{3.5}$$

The resulting distance matrix $D^{(t)} \in \mathbb{R}^{N \times N}$ is transformed into a similarity-based adjacency via a Gaussian kernel as shown in Eq. 3.6:

$$\mathbf{A}_{\text{cyc}}[t] = \exp\left(-\frac{(D^{(t)})^2}{2\sigma^2}\right) \tag{3.6}$$

where σ is the standard deviation computed across all pairwise distances over time. This produces a time-varying adjacency tensor $\mathbf{A}_{\text{cyc}} \in \mathbb{R}^{T \times N \times N}$ that encodes cyclic correlations at each temporal step.

3.2.2.1.3 Critical Node-Aware Graph Construction Irregular events such as accidents, roadwork, or sudden congestion induce localised disruptions that propagate through the network. These events are characterised by rapid fluctuations at specific nodes, which act as critical points influencing surrounding regions. The Critical Node-

Aware Graph identifies and emphasises interactions among such high-activity nodes. We quantify temporal activity at each node i through a fluctuation score as shown in Eq. 3.7:

$$f_i = \frac{1}{T-1} \sum_{t=0}^{T-2} |\mathbf{X}_{i,t+1} - \mathbf{X}_{i,t}| \tag{3.7}$$

measuring the average magnitude of temporal changes. Nodes exhibiting the top-K highest fluctuation scores are designated as critical. We construct a binary mask $\mathbf{M} \in \{0,1\}^{N \times N}$ that selectively retains edges between critical nodes as shown in Eq. 3.8:

$$M_{ij} = \begin{cases} 1 & \text{if } i, j \in \arg\,\text{top-K}(\mathbf{f}) \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

which is applied via Hadamard Product (element-wise multiplication) to an existing adjacency (e.g., physical topology or $\mathbf{A}_{\text{dyn}}$): $\mathbf{A}_{\text{mask}} = \mathbf{A} \odot \mathbf{M}$. This produces a subgraph that focuses computational resources on the most dynamic regions of the network.

3.2.2.2 Multi-View Graph Correlation Network

Having constructed three complementary adjacency matrices, $\mathbf{A}_{\text{dyn}}$ for adaptive correlations, $\mathbf{A}_{\text{cyc}}$ for cyclic patterns, and $\mathbf{A}_{\text{mask}}$ for critical node interactions, we perform spatial feature propagation through a unified Graph Convolutional Network (GCN) architecture. Despite the distinct semantic meanings of each adjacency, they share a common propagation mechanism that aggregates neighbourhood information according to the specified graph structure. The layer-wise update rule for node embeddings is formulated as the GCN operation in Eq. 3.9:

$$\mathbf{H}^{(l+1)} = \sigma\Big(\text{LayerNorm}\big(\mathbf{A}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\big)\Big) \tag{3.9}$$

where $\mathbf{H}^{(l)} \in \mathbb{R}^{N \times d_l}$ denotes the node feature matrix at layer l, $\mathbf{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$ is the trainable weight matrix, σ represents the ReLU activation, and $\mathbf{A}$ can be instantiated as any of the three adjacency matrices. Layer normalization stabilizes training by normalizing features across the embedding dimension.

By applying this propagation rule separately with each adjacency matrix, we obtain three distinct spatial embeddings: $\mathbf{H}_{\text{dyn}}$, $\mathbf{H}_{\text{cyc}}$, and $\mathbf{H}_{\text{mask}}$, each capturing a unique facet of the spatio-temporal structure. These multi-view representations are subsequently fused to provide a comprehensive understanding of traffic dynamics, enabling the model to handle adaptive, periodic, and irregular patterns simultaneously within a unified hierarchical framework. Moreover, as $\mathbf{H}_{\text{dyn}}$ represents the changing correlation impact on traffic flow and $\mathbf{H}_{\text{cyc}}$ represents the historical view, the net flow can be represented as the combination of both. Hence, we employ the Hadamard Product ($\odot$) for element-wise multiplication of $\mathbf{H}_{\text{dyn}}$ and $\mathbf{H}_{\text{cyc}}$ to create the combined Dynamic-Cyclic flow embedding $\mathbf{H}_{\text{dc}}$ as shown in Eq. 3.10:

$$\mathbf{H}_{\text{dc}} = \mathbf{H}_{\text{dyn}} \odot \mathbf{H}_{\text{cyc}} \tag{3.10}$$

The resulting embedding $\mathbf{H}_{\text{dc}}$ is then passed further with $\mathbf{H}_{\text{mask}}$.

3.2.3 Dual-Stream Attention Network

The expansion of the receptive field using retrieval (in Block A) can also be used to learn long receptive context using embeddings, and thus we used three dedicated encoders: a Time Series Encoder to learn the sequential embeddings, a Periodic Temporal Encoder to model time-step and periodic patterns, and a Topological Structure Encoder to learn the structural embeddings of the underlying network. Furthermore, these embeddings should learn useful features from the long context during training and provide them during inference. To achieve this, we used two independent attention modules, applied to the spatial and temporal dimensions, respectively. Spatial dependencies are captured using multi-head self-attention across nodes, while temporal dependencies are modelled via attention across the historical sequence. We encode

traffic periodicity and topological structure through data-driven spatio-temporal embedding learning. This approach facilitates a unified representation by simultaneously disentangling spatial and temporal dependencies. These learned representations now need to be combined with the output features of Block B (i.e., $\mathbf{H}_{\text{dc}}$ and $\mathbf{H}_{\text{mask}}$). For this unification, we stack the two outputs of the spatial and temporal attention modules, $\mathbf{H}_{\text{spatial}}$ and $\mathbf{H}_{\text{temporal}}$, and use a Gated Linear Unit [22] to fuse them with the stacked output features of Block B.

3.2.4 Integrated Prediction Network

The Unified Features, as an output of Block C, are refined further by passing them to a Transformer Encoder Layer, achieving global context refinement through self-attention across all fused features. Further, these rich features are passed through a Multilayer Perceptron (MLP) for final linear and nonlinear adjustments needed in high dimensional feature space. At last, we use a regression layer that projects the high-dimensional features into the desired output shape $\hat{\mathbf{X}} \in \mathbb{R}^{H \times N \times C}$ to produce the final predicted sequences for the next H time steps.

Chapter 4

Experimental Evaluations and Result

In this section, we present the details about the datasets, evaluation metrics, and compared methods. Subsequently, we also discuss the comparative analysis with respect to the state of the art methods, followed by the analysis of hyperparameters and performance gains through ablation studies.

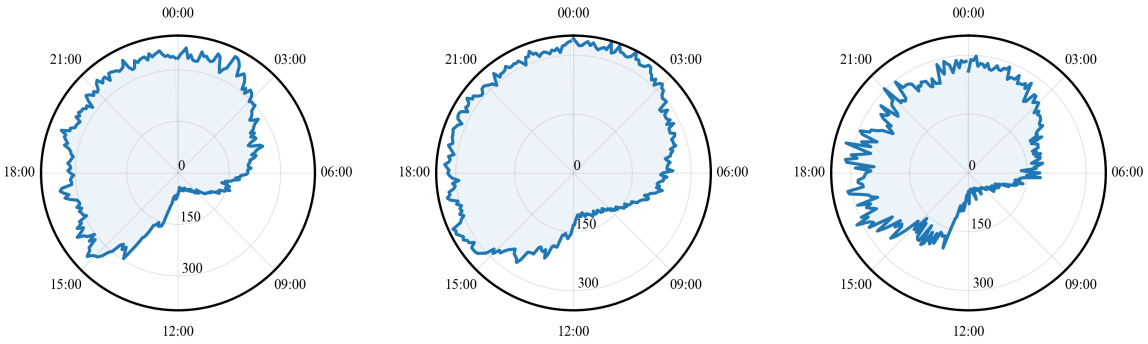
4.1 Datasets and Evaluation Metrics

To evaluate performance and generalisation, we conducted experiments on four primary traffic flow datasets, namely PEMS03, PEMS04, PEMS07, and PEMS08, and on two secondary traffic speed datasets, namely PEMS-BAY and METR-LA [4]. The PEMS datasets were collected by the California Transportation Agencies (CalTrans) Performance Measurement System (PeMS), while METR-LA was collected from loop detectors on the Los Angeles County road network. In all these datasets, the traffic data is aggregated into 5-minute intervals. A statistical summary presenting the number of sensor locations, total recorded timesteps, and the start and end dates of the traffic recordings is provided in Table 4.1.

Our comprehensive evaluation rigorously assesses the model's performance using established metrics, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE).

4.2 Exploratory Data Analysis (EDA)

We conducted EDA on various datasets used in this study. Here in this section, some insights from the PEMS08 traffic flow dataset to highlight its inter-sensor variability, periodic structure, and temporal heterogeneity.



(a) Node #15 (b) Node #40 (c) Node #150

Figure 4.1: Daily traffic flow trends on different sensors of the PEMS08 dataset.

Fig. 4.1 visualises the average diurnal traffic profile over a 24-hour cycle for multiple days. A consistent daily rhythm is observed, with traffic intensity peaking during late evening and early night hours, while reaching pronounced minima around midday. However, the amplitude and sharpness of these patterns vary across nodes, indicating non-stationary temporal dynamics of different sensors. Some nodes exhibit smoother transitions, whereas others show abrupt drops and recoveries, reflecting irregular congestions and inter-node disturbances.

Fig. 4.2 visualises the hourly mean distribution of traffic over the whole network of the PEMS08 dataset. The plots over the seven days of the week emphasise a strong

Table 4.1: Statistical summary of the benchmark datasets.

Dataset	Sensors	Timesteps	Type	Start	End
PEMS03	358	26,208	Flow	2018-09-01	2018-11-30
PEMS04	307	16,992	Flow	2018-01-01	2018-02-28
PEMS07	883	28,224	Flow	2017-05-01	2017-08-31
PEMS08	170	17,856	Flow	2016-07-01	2016-08-31
METR-LA	207	34,272	Speed	2012-03-01	2012-06-27
PEMS-BAY	325	52,116	Speed	2017-01-01	2017-06-30

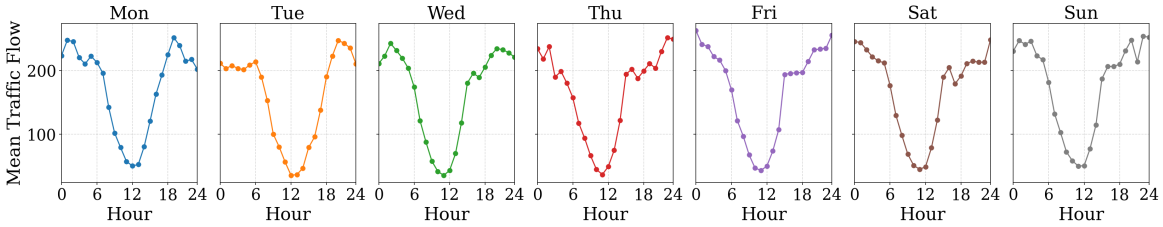


Figure 4.2: Weekly traffic flow trend on PEMS08 dataset.

periodic pattern along with some slight shift over the days of the week. While all days share a characteristic trough around midday, the depth and recovery rate differ, suggesting that both daily routines and weekday semantics influence traffic demand. Weekend patterns exhibit comparatively flatter temporal profiles, reflecting strong periodic regularity and reduced forecasting complexity.

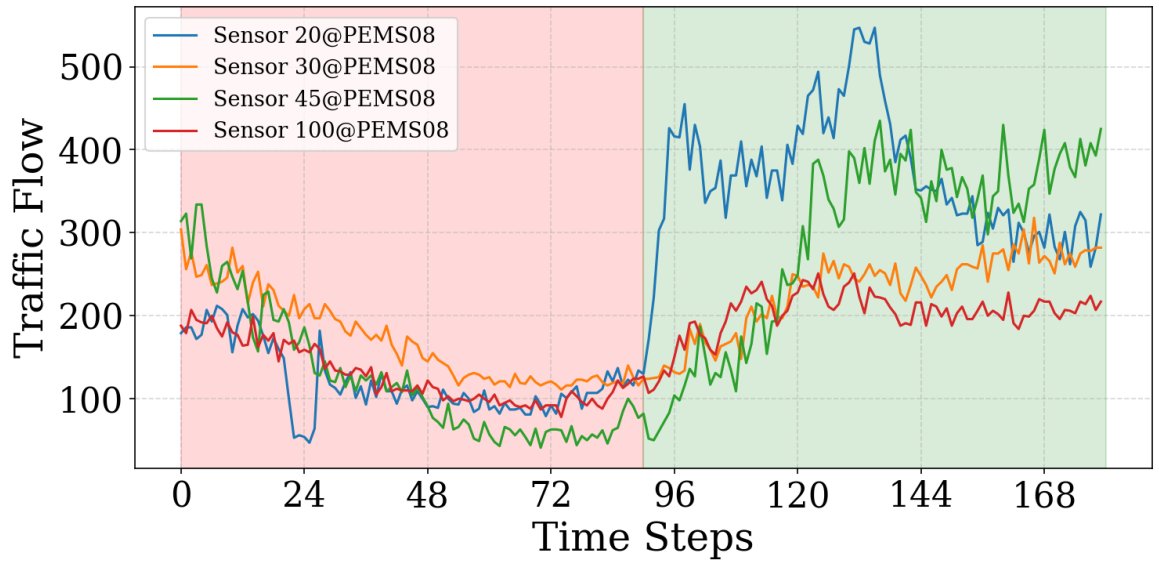


Figure 4.3: Spatio-Temporal illusions in Traffic flow of PEMS08 dataset.

Fig. 4.3 shows the multi-sensor temporal plot on the PEMS08 dataset. We plot sensors 20, 30, 40, and 45, which remain closely aligned up to the 90th timestep. At timestep 91, sensor 20 shows a sudden traffic surge and deviates from the others. Similarly, at timestep 120, sensor 45 also diverges from the remaining sensors. Meanwhile, sensors 30 and 100 remain aligned, despite others deviating. This demonstrates that similar historical traffic patterns may evolve differently. Such behaviour creates a spatio-temporal illusion, confusing prediction models.

These observations justify the need for adaptive receptive field expansion and multi-view spatial modelling, as static temporal windows or single-view graphs may fail to capture such diverse and evolving traffic patterns effectively.

4.3 Compared Methods

We compare our work with four types of existing state-of-the-art methods in the traffic flow prediction:

4.3.1 Classical & Statistical Methods

This group includes fundamental time-series baselines such as HA [3], predicting future values using the mean of historical data, ARIMA [3], a classical model for capturing linear temporal dependencies, VAR [8], that captures linear temporal dependencies across multiple time series, and SVR [8], a machine learning approach that uses kernel methods to model non-linear relationships.

4.3.2 Classic Spatio-Temporal GNN Methods

This group includes early Spatio-Temporal Graph Neural Networks (ST-GNNs) for joint spatial–temporal modeling. DCRNN [7] methods spatial dependencies via diffusion graph convolution within a recurrent framework, STGCN [5] adopts a fully convolutional architecture with graph and gated temporal convolutions, GWNet [6] integrates temporal convolution with an adaptive adjacency matrix, MTGNN [9] learns discrete graph structures using mix-hop propagation, AGCRN [2] infers adaptive graphs with node-specific parameters, and STEMGNN [12] jointly captures inter-series correlations and temporal dependencies in the spectral domain.

4.3.3 Adaptive Graph-Learning Methods

This group includes methods that improve spatio-temporal forecasting via dynamic correlation graph learning. GTS [10] learns probabilistic graph structures to

address non-stationarity, ST-Norm [18] stabilises feature distributions through spatial–temporal normalization, DGCRN [11] introduces dynamic graph convolution with time-varying adjacency matrices, HimetNet [23] leverages heterogeneity-aware meta-learning, and LightST [24] simplifies ST-GNN architectures for improved efficiency without sacrificing performance.

4.3.4 Transformer-Enhanced Spatio-Temporal Methods

This group comprises methods that employ Transformer or attention mechanisms for temporal modeling, often combined with GNNs or pre-training strategies. STID [13] adopts spatial and temporal identity embeddings with a simple feed-forward architecture, STEP [14] enhances a GNN backbone via unsupervised pre-training, STWave [15] disentangles representations using wavelet transforms with efficient graph attention, PDFormer [25] incorporates propagation-delay-aware Transformers, STAEformer [16] introduces spatio-temporal adaptive embeddings for effective Transformer forecasting, Starformer [26] fuses GNNs with Transformers through a series-core mechanism, STDMAE [19] proposes a decoupled masked autoencoder for spatio-temporal pre-training, STGAFormer [17] integrates gated spatio-temporal attention within GNNs, and MVSTDG [27] constructs multi-view graphs to capture spatio-temporal dependencies.

4.4 Experimental Settings

We conducted all experiments using the fair and scalable time series forecasting framework BasicTS [4]. As described in the problem definition, a historical observation window of length L is used to forecast the next H future steps. In our setup, we set $L = 12$ and $H = 12$. Since all datasets are sampled at five-minute intervals, the models use the previous 60 minutes of observations to predict traffic conditions over the subsequent 60 minutes.

For each dataset, we chronologically divided the time series into training, validation, and test sets using a ratio of 60:20:20, respectively. Z-score normalisation was

applied to the input sequences, with the mean and standard deviation calculated using the training set only and subsequently applied to the validation and test sets. This procedure prevents information from the validation and test sets from influencing the normalisation parameters. Training samples were generated using a sliding-window approach within the corresponding data partitions.

For model training, all methods were trained for 100 epochs using a batch size of 64. The Adam optimiser was used with an initial learning rate of 0.002 and a weight decay of 0.0001. A multi-step learning-rate scheduler was employed, reducing the learning rate by a factor of 0.5 at epochs 1, 50, and 80. The mean absolute error (MAE) was used as the training loss. For each model configuration, the checkpoint achieving the lowest validation MAE was selected for final evaluation. The test set was not used for model training or checkpoint selection.

For testing, the selected checkpoint was evaluated on the held-out test set without further parameter updates. The evaluation metrics included mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). Before calculating these metrics, the predictions and corresponding ground-truth values were transformed back to their original data scale. A small constant was added to the denominator when calculating MAPE to handle zero-valued observations. The reported metrics were aggregated over all 12 future prediction steps, corresponding to a one-hour forecasting horizon.

All baseline models were evaluated using the same chronological data split, input window, forecasting horizon, and evaluation metrics as the proposed method to ensure a consistent comparison. Each model configuration was run once for the main experimental results. Additional runs were conducted on PEMS03 and PEMS08 to check the consistency of the results; however, these repeated runs were not averaged into the main results tables.

All experiments were executed on a high-performance compute node equipped with dual-socket Intel Xeon Silver 4316 CPUs providing 80 threads, 512 GB RAM, and an NVIDIA A16 GPU cluster consisting of four GPUs with 16 GB memory each. This hardware configuration supported the training and evaluation of the proposed method

and baseline models under a consistent experimental environment.

4.5 Experimental Results

Table 4.2: Performance comparison on PEMS03 and PEMS04 using MAE, RMSE, and MAPE across baselines.

Model	PeMS03			PeMS04		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA [3]	33.30	56.35	32.84	40.51	61.11	25.72
ARIMA [3]	36.04	47.99	34.18	33.75	48.82	24.58
SVR [8]	28.11	26.94	44.62	28.09	44.86	19.34
VAR [8]	23.71	38.91	24.36	23.87	35.94	17.96
DCRNN [7]	15.51	27.13	15.59	19.59	31.32	13.56
STGCN [5]	15.59	27.59	15.67	19.69	31.47	13.63
GWNet [6]	15.54	27.10	15.57	19.32	31.16	13.55
STEMGNN [12]	15.31	26.91	15.79	21.59	33.48	16.03
MTGNN [9]	15.21	26.58	15.63	19.39	31.98	13.22
AGCRN [2]	15.27	26.70	15.92	18.97	31.01	13.43
GTS [10]	15.35	26.05	14.31	18.31	29.83	12.64
ST-Norm [18]	15.18	25.92	14.28	19.03	31.15	12.90
STID [13]	15.36	27.50	16.39	18.45	29.96	12.58
STEP [14]	14.21	24.57	14.48	18.23	29.89	12.46
STWave [15]	14.66	25.69	15.10	18.32	30.10	12.30
PDFormer [25]	15.31	25.87	15.19	18.42	30.09	12.55
STAEformer [16]	15.51	28.06	15.23	18.30	30.26	12.03
Starformer [26]	19.42	30.88	12.51	18.68	30.62	12.83
DGCRN [11]	14.70	26.25	15.16	18.93	30.74	13.01
HimetNet [23]	15.55	26.67	15.55	18.21	30.00	12.05
STDMAE [19]	14.10	24.51	14.03	18.17	30.11	12.05
STGAFormer [17]	14.51	24.96	14.78	18.20	29.97	11.99
LightST [24]	14.66	24.81	15.06	18.11	29.71	11.88
MVSTDG [27]	14.54	25.12	14.85	18.49	30.06	12.18
STGLRNet	14.00	23.01	14.74	17.29	28.36	11.79

4.5.1 Performance Comparison on Traffic Flow Datasets

We compare our proposed method with the state-of-the-art methods as shown in Table 4.2 and Table 4.3. We used four benchmark traffic flow datasets, namely PEMS03, PEMS04, PEMS07, and PEMS08. We use a range of baselines which covers

Table 4.3: Performance comparison on PEMS07 and PEMS08 using MAE, RMSE, and MAPE across baselines.

Model	PeMS07			PeMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA [3]	45.06	65.25	24.25	34.91	59.53	26.98
ARIMA [3]	37.68	59.16	20.32	30.72	44.50	22.87
SVR [8]	32.67	51.74	14.79	24.39	36.74	14.59
VAR [8]	50.28	76.08	33.73	21.42	30.49	13.20
DCRNN [7]	21.12	34.07	9.22	15.19	24.12	10.58
STGCN [5]	21.22	34.24	9.27	15.27	24.24	10.24
GWNet [6]	20.41	33.37	8.99	15.17	23.32	10.18
STEMGNN [12]	22.25	36.44	9.18	15.84	25.43	10.89
MTGNN [9]	21.04	33.86	9.06	15.31	24.44	10.69
AGCRN [2]	20.19	34.54	9.10	15.35	24.13	10.05
GTS [10]	19.53	32.97	8.72	14.15	23.38	9.72
ST-Norm [18]	20.52	34.49	8.67	15.43	24.89	9.69
STID [13]	19.6	32.71	8.29	14.29	23.35	9.32
STEP [14]	19.61	32.27	8.15	14.03	23.64	9.53
STWave [15]	19.52	32.89	8.30	13.83	23.81	9.50
PDFormer [25]	19.78	33.02	8.79	15.44	23.52	9.96
STAEformer [16]	19.26	32.96	8.31	13.51	23.55	8.87
Starformer [26]	20.38	33.06	9.12	13.98	23.43	9.45
DGCRN [11]	20.21	32.17	10.37	14.25	23.63	9.31
HimetNet [23]	19.29	32.88	8.06	13.62	23.31	9.02
STDMAE [19]	19.16	31.26	7.85	13.51	22.98	8.97
STGAFormer [17]	19.66	32.65	8.51	13.07	22.45	8.91
LightST [24]	19.27	32.46	8.07	13.45	23.05	8.89
MVSTDG [27]	19.26	32.30	8.12	13.39	22.74	9.07
STGLRNet	18.62	30.85	8.25	12.97	21.63	8.78

classical methods such as ARIMA, recurrent graph neural networks such as DCRNN, adaptive graph learning approaches such as GWNet, and recent Transformer-based architectures such as STGAFormer. Our model demonstrates superior performance across all four datasets, achieving the lowest MAE, RMSE, and MAPE in the majority of cases. Specifically, on PEMS08, we achieve an MAE of 12.97, RMSE of 21.63, and MAPE of 8.78%, representing improvements of 3.13%, 3.82%, and 1.3% on the respective metrics over the previous best-performing method STGAFormer. Similar performance gains are observed on PEMS03, PEMS04, and PEMS07. These consistent improvements across diverse datasets validate the effectiveness of our retrieval-

augmented approach and multi-view hierarchical pattern recognition framework.
Our method outperforms existing approaches due to its ability of Receptive Field Retrieval which enables to increase receptive field beyond the fixed input window. Using similarity search relevant past traffic sequences are extracted from the previously observed history, embedded in our vector database. Along with that our Hierarchical Pattern Recognition Network handles spatial relationships more realistically by building three complementary views.
Fig. 4.5 shows horizon wise prediction error including MAE, RMSE, and MAPE. Magnitude of errors is directly proportional to number of prediction horizons. In our method the rise in errors over the horizons is comparatively small. Our MAE goes from 11.68 at horizon 1 to 13.79 at horizon 12 similarly RMSE from 18.67 at horizon 1 to 23.21 at horizon 12, and MAPE from 7.71% at horizon 1 to 9.52% at horizon 12. That kind of stability is rare in long-range forecasting and it's exactly what's needed for real applications like dynamic routing or anticipating congestion before it happens.
Fig. 4.4 gives a visual sense of performance. Over a continuous stretch of time on PEMS08, our method tracks the ground truth closely, even during rush hour or sudden drops in flow. In contrast, five strong baselines including STGCN, GWNet, AGCRN, STID, and STWave our method consistently capture the traffic trend with stability.

4.5.2 Comparison of Performance on Traffic Speed Datasets

We assessed the generalizability of `STGLRNet` beyond traffic flow forecasting by evaluating its performance on two standard traffic speed benchmarks: METR-LA and PEMS-BAY. Table 4.4 presents multi-horizon predictions at 15-minute (@3), 30-minute (@6), and 60-minute (@12) intervals, compared against five strong baselines, i.e., GWNet, STGCN, STID, STWave and STNorm.

On the PEMS-BAY dataset, the STGLRNet model achieved the best performance across all tested horizons and metrics. Specifically, for MAE, the model attained values of 1.24, 1.53, and 1.75 at the 15-minute, 30-minute, and 60-minute horizons, respectively. These results represent significant improvements, including gains of 7.4% over

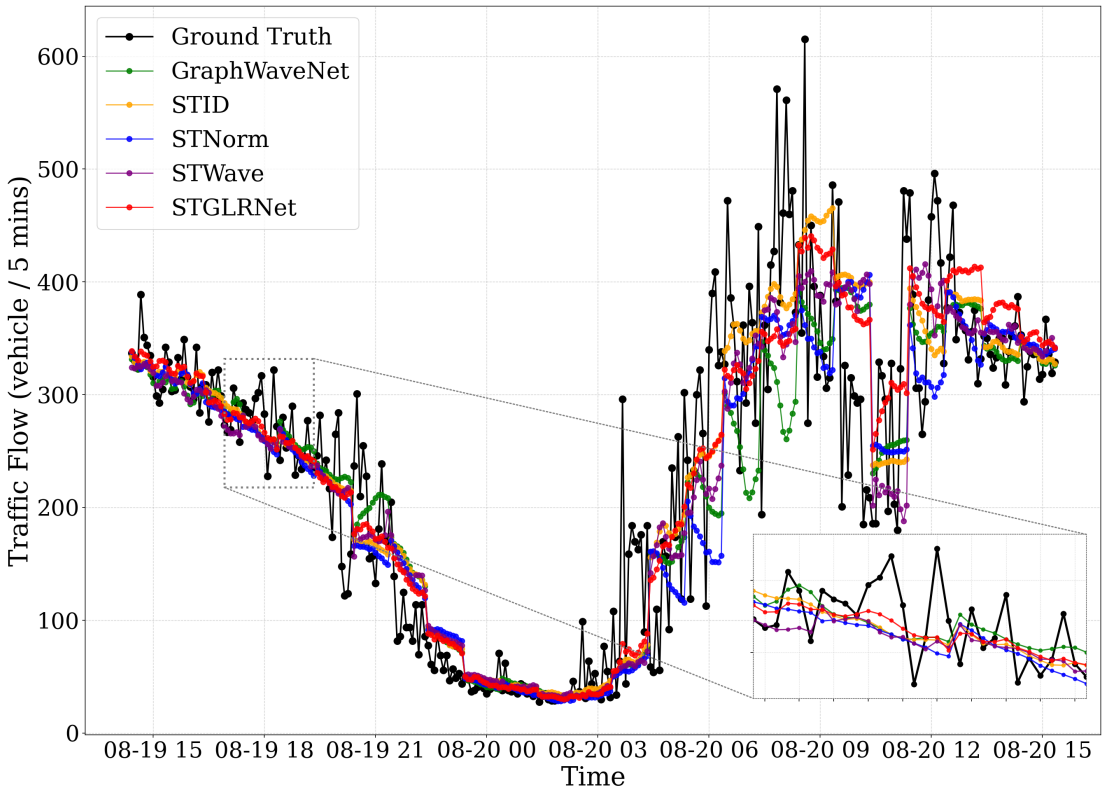


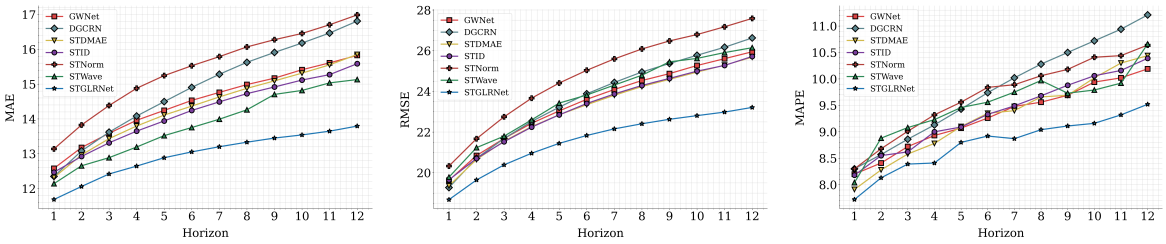
Figure 4.4: Predictions V/s Ground Truth over our proposed method with five baselines for 300 steps on Sensor Number 150 of the PEMS08 Dataset.

the second-best method, i.e., STID, at the longest 60-minute horizon. This demonstrates the effectiveness of the retrieval-augmented architecture in transferring to speed prediction tasks, where temporal and spatial dynamics differ from flow data.

For the METR-LA dataset, STGLRNet achieved competitive performance, ranking among the top methods across all three horizons with MAE values of 2.77, 3.14, and 3.50. While some baselines showed marginally better performance at shorter horizons, our proposed model demonstrated superior accuracy at the long 60-minute horizon (@12). This observation suggests that the retrieval-based context augmentation, which dynamically expands the receptive field, becomes increasingly valuable as the prediction window extends and uncertainty accumulates. The slightly smaller performance margin on METR-LA, compared to PEMS-BAY, may be linked to specific characteristics of the Los Angeles network, such as its higher spatial density of sensors, which reduces the relative importance of long-range historical context retrieval.

Table 4.4: Performance comparison of methods on METR-LA and PEMS-BAY datasets.

Dataset	Model	MAE			RMSE			MAPE		
		@3	@6	@12	@3	@6	@12	@3	@6	@12
METR-LA	GWNet	**2.69**	**3.08**	3.51	**5.15**	**6.20**	**7.28**	**6.99**	**8.47**	**9.96**
	STGCN	2.77	3.17	3.65	5.33	6.37	7.50	7.15	8.69	10.50
	STID	2.80	3.17	3.54	5.52	6.57	7.52	7.61	9.22	10.79
	STWave	2.84	3.20	3.56	5.58	6.56	7.46	7.67	9.22	10.66
	STNorm	2.83	3.22	3.59	5.58	6.69	7.59	7.48	9.01	10.49
	STGLRNet	2.77	3.14	**3.50**	5.43	6.44	7.32	7.36	8.84	10.33
PEMS-BAY	GWNet	1.30	1.63	1.99	2.73	3.73	4.60	2.71	3.73	4.71
	STGCN	1.42	1.73	2.04	2.95	3.85	4.64	3.05	3.87	4.71
	STID	1.30	1.61	1.89	2.76	3.68	4.36	2.75	3.68	4.44
	STWave	1.32	1.67	1.90	2.81	3.76	4.35	2.72	3.77	4.47
	STNorm	1.33	1.66	1.94	2.85	3.81	4.48	2.79	3.70	4.56
	STGLRNet	**1.24**	**1.53**	**1.75**	**2.67**	**3.56**	**4.14**	**2.64**	**3.48**	**4.16**



(a) Horizon Wise MAE (b) Horizon Wise RMSE (c) Horizon Wise MAPE

Figure 4.5: Horizon-wise evaluation of MAE, RMSE, and MAPE on the PEMS08 dataset.

4.5.3 Performance Gain Analysis

To systematically evaluate the contribution of each architectural component, we conduct hierarchical ablation studies organised by the three major modules: Hierarchical Pattern Recognition Network (HPRN), Dual-Stream Attention Network (DSAN), and Memory-Enhanced Feature Extraction Unit (MFEU). Table 4.5 summarises the results on PEMS08.

4.5.3.1 Impact of Hierarchical Pattern Recognition Network

We built the HPRN module around three spatial graphs because real traffic doesn't obey a single set of rules. When we remove the whole module, MAE rises by 6.8%

Table 4.5: Hierarchical ablation study of STGLRNet.

Component	Variant	MAE	RMSE	MAPE (%)
HPRN	w/o ACG	13.10	21.81	9.00
	w/o CPG	13.08	21.79	8.99
	w/o CNG	13.18	21.97	9.06
	w/o (ACG+CPG+CNG)	13.85	23.12	9.11
DSAN	w/o Encoders	13.16	21.89	9.04
	w/o Attention	13.11	21.71	8.96
	w/o (Encoders+Attention)	13.55	22.80	9.01
MFEU	w/o Complete MFEU	13.59	22.88	8.93
	STGLRNet	12.97	21.63	8.78

HPRN: Hierarchical Pattern Recognition Network; DSAN: Dual-Stream Attention Network; MFEU: Memory-Enhanced Feature Extraction Unit; ACG: Adaptive Correlation Graph; CPG: Cyclic Preparatory Graph; CNG: Critical Node-Aware Graph.

indicating that modeling space from multiple angles matters. Removing each graph one at a time shows that the critical-node-aware adjacency has the largest impact as it captures those nodes which are responsible for bigger fluctuations and irregularity in traffic patterns. Similar performance drops were observed on the removal of cyclic preparatory graph and adaptive correlation graph. This suggests that using multi-view adjacency generates better features for further graph correlation networks.

4.5.3.2 Impact of Dual-Stream Attention Network

On removing DSAN module it degrades performance by 4.28%, we ablated its two components separately. Taking out the attention mechanism alone reduces performance by 1.06%. Removing the embedded encoders causes a larger performance drop 1.44% which tells us these embedding encoders capture useful data inherent spatio-temporal structure that raw inputs miss.

4.5.3.3 Impact of Memory-Enhanced Feature Extraction Unit

Absence of MFEU resulted in a 4.8% degradation in performance. This is due to the fact that MFEU provides a broader receptive field for the model by retrieving

relevant historical contexts from the temporal database. Subsequently this allows the model to capture long-range context that would not be accessible within a fixed-length input window.

This ablation studies shows that each module has its importance and removal of same leads to degradation in the overall performance. The HPRN grounds predictions in realistic spatial structure. The DSAN stabilizes temporal modeling by balancing learned features with attention-based refinement. And the MFEU extends the model's memory far beyond the input window, enabling it to respond to complex, low-frequency dynamics. When any one piece is missing, performance suffers noticeably proving that the architecture works as an integrated system, not just a collection of parts.

4.5.4 Qualitative Analysis and Pattern Visualization

To provide a deeper insight into the predictive capability of the model, Fig. 4.6 visualizes the prediction versus ground truth over 3000 steps of the PEMS08 dataset. This long-horizon visualisation demonstrates the model's ability to maintain prediction accuracy over extended periods without significant drift or error accumulation. Equivalent visualisations for the METR-LA and PEMS-BAY speed datasets are provided in the Supplementary Material. Also, the predicted traffic flow on PEMS08

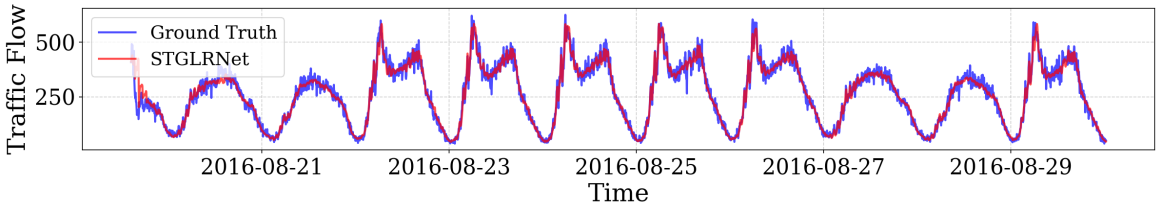


Figure 4.6: Predicted vs. Ground Truth comparison for 3000-step prediction on the PEMS08 dataset.

closely aligns with ground truth across multiple daily cycles, successfully capturing morning and evening rush hour peaks as well as overnight low-traffic periods. This shows our method exhibits strong performance in tracking abrupt transitions between congested and free-flow states, which are critical for proactive traffic management.

Fig. 4.7 visualises the dynamically changing spatial correlations changing over time

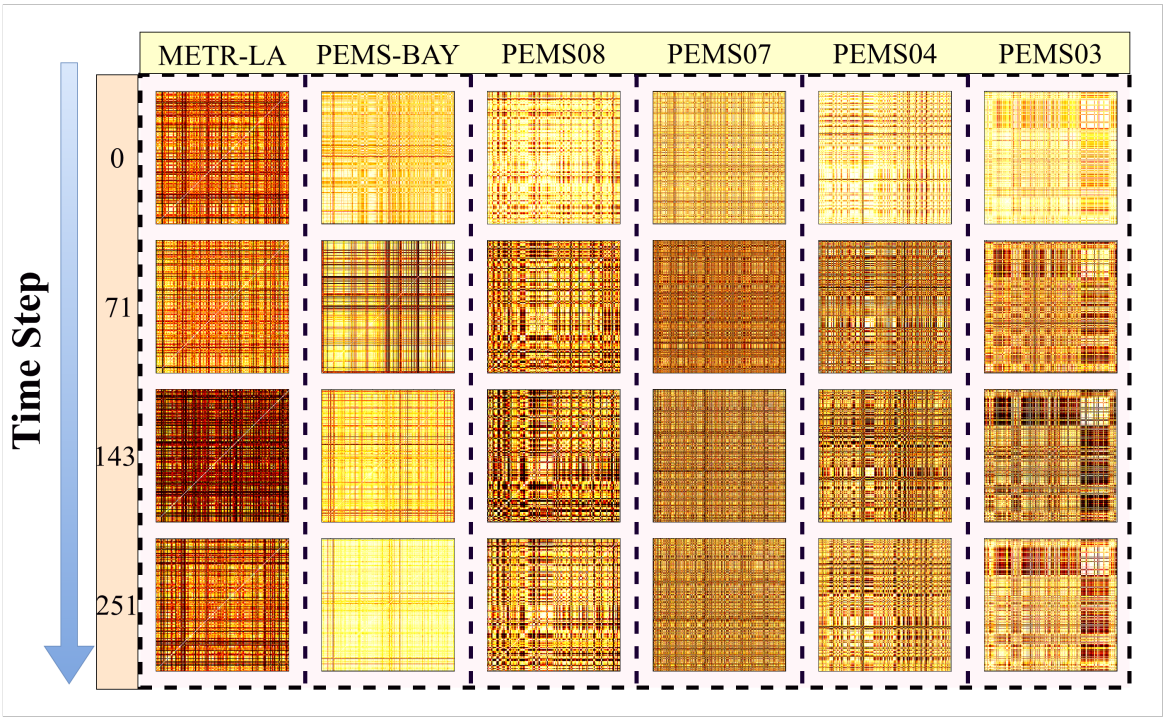


Figure 4.7: Dynamically Changing Spatio-Temporal Patterns.

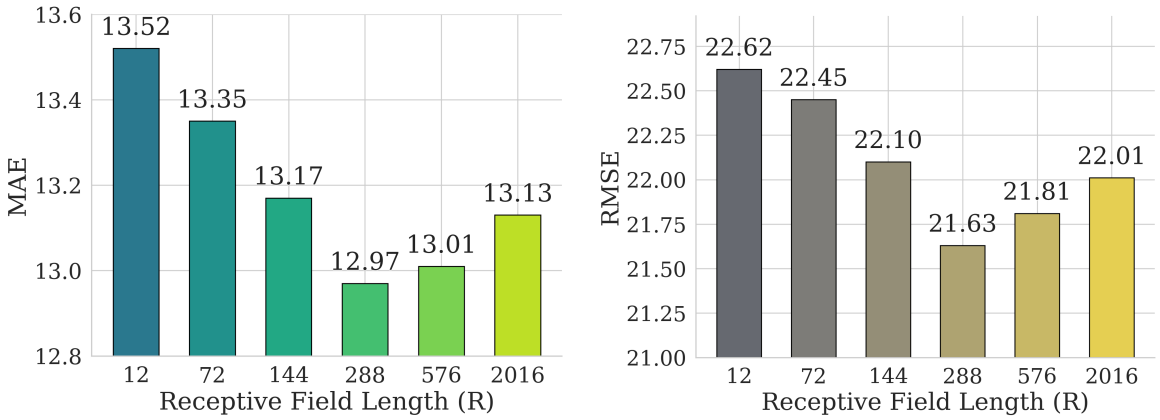
learned through our method across different time steps of a day, such as 0, 71, 143, 251. The heatmaps illustrate the inter-node correlation in traffic patterns in the form of an NXN correlation matrix. Strongly correlated patterns are characterised by more saturated means and vice versa. This visualisation validates our hypothesis that traffic dependencies are inherently dynamic and require adaptive modelling mechanisms rather than fixed graph structures. The temporal evolution of these patterns also suggests that the model successfully learns to distinguish between different traffic regimes and adjusts its spatial reasoning accordingly.

4.5.5 Hyperparameter Sensitivity Analysis

In this section, we will discuss how model performance changes when we adjust our two primary hyperparameters:

4.5.5.1 Receptive Field Length

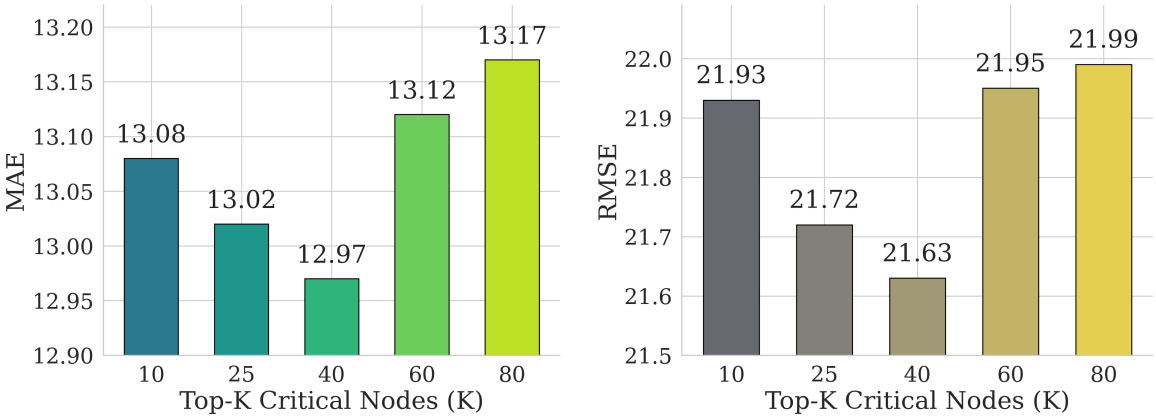
Fig. 4.8 shows how model performance changes as we vary the amount of historical context retrieved. Both MAE and RMSE improve as the receptive field grows, reaching their best values at around 288 time steps. At this point, MAE drops from 13.52 to 12.97 and RMSE from 22.62 to 21.63. The gain comes from accessing longer traffic context from the current traffic conditions. But going beyond 288 steps doesn't help. Performance plateaus and then slightly worsens. The reason is intuitive as very old sequences may look similar in embedding space, but they often reflect different road conditions and becomes less and less relavant for current traffic state.



(a) MAE (b) RMSE

Figure 4.8: Sensitivity of performance metrics (MAE and RMSE) to R (i.e., receptive field length) on the PEMS08 dataset.

The optimal receptive field retriever significantly decreased the MAE from 13.52 to 12.97 and RMSE values from 22.62 to 21.63. This is because it captures extended traffic patterns, providing a sufficiently unique context for pattern matching via FAISS retrieval, and avoids incorporating excessively distant patterns that may have limited relevance to current conditions. The relative degradation in performance beyond the optimal point suggests that traffic conditions become progressively less influential as time progresses and hence act more like noise, making predictions worse.



(a) MAE (b) RMSE

Figure 4.9: Sensitivity of performance metrics (MAE and RMSE) to k (i.e., number of critical nodes) on the PEMS08 dataset.

4.5.5.2 Number of Critical Nodes

Fig. 4.9 analyses the impact of k, the number of critical nodes selected for the CNG adjacency matrix. Performance improves as k increases from 10 to 40 nodes, then degrades. This trend indicates that including too few critical nodes fails to capture sufficient irregular patterns, while including too many dilutes the focus on true anomalies.

We notice that the optimal range of critical nodes in PEMS08 is 40 i.e., about 23% of the 170 of the total nodes. This suggests that traffic irregularities typically present at a relatively small fraction of the network. It also validates the use of masked adjacency used in irregular pattern mining network.

Chapter 5

Discussion

This chapter presents a broader discussion of the proposed STGLRNet framework, its architectural strengths, remaining limitations, and future research directions motivated by subsequent investigations. While STGLRNet demonstrated strong forecasting performance across multiple benchmark datasets through retrieval-augmented receptive field expansion and multi-view spatio-temporal graph learning, several important research questions remain open. These limitations motivated the exploration of alternative paradigms in subsequent works, namely LTFNet and GATP, which attempt to address some of these challenges from fundamentally different perspectives.

5.1 Temporal Database Scalability and Index Management

To ensure the temporal database remains bounded during continuous deployment, we adopt a fixed-capacity forgetting mechanism. The index retains at most $\mathcal{C} = N \times T_{\text{window}}$ entries, where $T_{\text{window}} = 2{,}016$ timesteps corresponds to one week of observations at 5-minute resolution, precisely the dominant periodicity in traffic data [4]. When new observations are indexed, entries exceeding this horizon are evicted via FAISS's `remove_ids()` interface, maintaining $\mathcal{O}(N \cdot T_{\text{window}})$ memory complexity independent of deployment duration. This bound is practically modest: for PEMS08 ($N = 170$), the maximum index size is 342,720 vectors of dimension d'. For

memory-constrained systems, a sliding window with exponential temporal decay over retrieval scores offers a softer alternative:

$$\tilde{s}(t_i) = s(t_i) \cdot \exp(-\lambda \cdot (t - t_i)) \tag{5.1}$$

where $\lambda \to 0$ recovers the current hard-boundary behaviour, which we defer to future work. As the proposed architecture employs a fixed-capacity forgetting mechanism with temporal decay, it remains inherently robust to distribution shift over time.

5.2 Scalability and Deployment Considerations

STGLRNet comprises approximately 1.79 M trainable parameters, remaining lightweight relative to its multi-module architecture. On PEMS08 (N=170), the model trains in 327.06 s per epoch and completes batch inference (batch size 64) in 81.90 s, yielding a per-sample inference time of approximately 1.28 s. This is substantially faster than DGCRN [11] (302.54 s) and DSTRFormer [28] (130.40 s) at inference, and more efficient than STWave [15] (389.44 s), DGCRN [11] (826.54 s), DSTRFormer (2707.01 s), and Koopa [29] (597.76 s) in per-epoch training time. The efficiency gains stem from three design choices: (i) FAISS retrieval operates in sub-linear time, avoiding the quadratic cost of extending the attention window; (ii) the DTW-based cyclic adjacency is precomputed offline with zero runtime overhead; and (iii) the dual-stream attention disentangles spatial and temporal dimensions, halving the effective sequence length per stream. These properties position STGLRNet for deployment in real-time intelligent transportation systems, including adaptive signal control, dynamic route guidance, and congestion-aware fleet management, where both accuracy and latency are operationally critical.

5.3 Limitations and Future Work

Despite achieving competitive forecasting performance across multiple benchmark datasets, several limitations of STGLRNet remain open for future investigation.

First, the current memory retrieval formulation relies on sequence-level similarity search and retrieval-based receptive field expansion. While effective in alleviating fixed receptive field limitations, the framework still depends on explicit historical sequence matching. Future work may explore the integration of Test Time Training (TTT), adaptive neural memory systems, and gated memory architectures to enable more efficient, scalable, and continuously evolving retrieval mechanisms. Such approaches may further improve long-range dependency modeling while reducing retrieval overhead.

Second, although the proposed multi-view adjacency construction successfully captures heterogeneous spatial correlations, the interpretability and relative contribution of individual graph views remain insufficiently explored. In particular, a deeper investigation into adaptive, cyclic, and critical-node-aware adjacency matrices could provide valuable insights into which spatial relationships are most influential under varying traffic conditions. Future studies may also investigate explainable graph learning strategies and dynamic view selection mechanisms for improving interpretability and robustness.

Third, the current framework prioritizes forecasting accuracy over computational efficiency. While the architecture demonstrates competitive performance, real-world intelligent transportation systems often require ultra-low latency inference under resource-constrained deployment environments. Future work should therefore investigate lightweight optimization techniques such as model pruning, quantization, low-rank adaptation, and efficient attention approximations to better balance predictive performance and computational cost.

Finally, although STGLRNet effectively models complex spatio-temporal dependencies through retrieval-augmented sequence learning, it fundamentally operates within a discrete sequence-to-sequence forecasting paradigm. This raises broader ques-

tions regarding whether traffic forecasting can instead be formulated through continuous latent dynamical systems or transferable temporal representations independent of graph topology. These observations motivated subsequent investigations discussed in the following sections.

5.4 Beyond Retrieval-Based Forecasting: Continuous Latent Dynamics

While STGLRNet addresses the limitations of fixed receptive fields through retrieval-augmented context expansion, the framework still fundamentally operates in a discrete sequence-mapping paradigm. This observation motivated subsequent exploration into continuous latent dynamical modeling, resulting in the development of Latent Traffic Field Networks for Spatio-Temporal Traffic Forecasting (LTFNet).

LTFNet reformulates traffic forecasting as the evolution of a continuous latent traffic field governed through graph-constrained neural partial differential equations. Instead of retrieving historical trajectories, the model attempts to learn the underlying continuous spatio-temporal dynamics responsible for traffic evolution. This formulation shifts the focus from explicit sequence matching toward latent dynamical system modeling, enabling a more compact and physically grounded representation of traffic behavior.

In contrast to STGLRNet, which relies on memory-enhanced retrieval and multi-view graph reasoning, LTFNet introduces latent field evolution through diffusion operators, neural PDE dynamics, and continuous-time state-space refinement. The framework treats traffic observations as samples from an evolving continuous latent field distributed over the traffic network. Spatial propagation is modeled through graph Laplacian diffusion, while nonlinear traffic interactions are captured through neural operators. Furthermore, temporal dependencies are refined using structured state-space dynamics.

The motivation behind this direction arises from the observation that many existing

traffic forecasting methods, including retrieval-based approaches, attempt to directly learn large numbers of possible input-output mappings between historical and future sequences. Such formulations may become increasingly complex and computationally expensive as forecasting horizons and contextual variability increase. LTFNet attempts to overcome this challenge by modeling the underlying continuous dynamics governing traffic evolution instead of memorizing discrete sequence relationships.

Additionally, the latent field formulation introduces several desirable properties. First, it provides a physically interpretable perspective by linking traffic propagation to diffusion processes over road networks. Second, the continuous evolution framework enables smoother long-range temporal modeling. Third, the architecture remains comparatively lightweight while maintaining competitive forecasting performance. Consequently, LTFNet represents an alternative research direction focused on dynamical system modeling and continuous latent representations rather than retrieval-augmented sequence learning.

5.5 Towards Transferable and Topology-Agnostic Forecasting

Another important limitation observed in existing traffic forecasting systems, including STGLRNet, is their dependence on city-specific graph structures. Most spatio-temporal graph neural networks tightly couple temporal representation learning with graph topology, limiting their ability to generalize across unseen urban environments. This motivated the development of Graph-Agnostic Temporal Pretraining for Multi-City Traffic Forecasting (GATP).

Traditional graph-based forecasting models generally require complete retraining whenever deployed on a new city or road network topology. Since the learned temporal representations become strongly entangled with the graph structure during training, transferring knowledge across cities becomes difficult. This issue, referred to as topology-lock, creates major challenges for scalable deployment in real-world intelli-

gent transportation systems.

Unlike STGLRNet, which jointly learns spatial and temporal dependencies within a unified graph-based framework, GATP explicitly decouples temporal representation learning from graph reasoning. The framework introduces a graph-agnostic temporal pretraining strategy in which universal temporal dynamics are learned independently from any specific road topology. A lightweight graph conditioning module is subsequently used to adapt the pretrained temporal representations to unseen city networks during deployment.

The central idea behind GATP is based on the observation that many traffic dynamics, such as morning congestion peaks, evening traffic transitions, daily periodicity, and weekly temporal cycles, are shared across cities irrespective of road topology. Therefore, instead of repeatedly learning these temporal behaviors separately for every city, GATP attempts to learn generalized temporal representations from a large multi-city corpus through node-wise spatial disentanglement.

This direction addresses the broader challenge of cross-city transferability and scalable deployment of intelligent transportation systems. By separating temporal dynamics from topology-specific reasoning, GATP attempts to overcome the topology-lock problem inherent in many existing graph-based forecasting architectures. Furthermore, the use of frozen temporal encoders and lightweight graph conditioning modules significantly reduces retraining costs and improves adaptability to unseen traffic networks. This work has been published in *Physica A: Statistical Mechanics and its Applications* and is available online at doi:10.1016/j.physa.2026.132143.

Chapter 6

Conclusion

In this work, we address three fundamental challenges in traffic flow forecasting: a fixed context window and a limited graph view and learning spatio-temporal entangled dependencies. We propose STGLRNet, an adaptive spatio-temporal framework that dynamically expands receptive fields through retrieval-based context augmentation. The model integrates a hierarchical GNN that leverages multi-view correlation matrices to capture complex spatial dependencies. Additionally, dual-stream attention with learnable embedding learns periodic temporal patterns and network topology effectively. Extensive evaluations on six benchmark datasets demonstrate that STGLRNet achieves consistently superior and more stable long-horizon prediction performance.

Bibliography

[1] Y. Zheng, L. Capra, O. Wolfson, H. Yang, Urban computing: Concepts, methodologies, and applications, ACM Trans. Intell. Syst. Technol. 5 (3) (Sep. 2014). doi:10.1145/2629592.
URL https://doi.org/10.1145/2629592

[2] L. Bai, L. Yao, C. Li, X. Wang, C. Wang, Adaptive graph convolutional recurrent network for traffic forecasting, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020.
URL https://dl.acm.org/doi/abs/10.5555/3495724.3497218

[3] Z. Fang, Q. Long, G. Song, K. Xie, Spatial-temporal graph ode networks for traffic flow forecasting, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery &; Data Mining, KDD '21, ACM, 2021, p. 364–373. doi:10.1145/3447548.3467430.
URL http://dx.doi.org/10.1145/3447548.3467430

[4] Z. Shao, F. Wang, Y. Xu, W. Wei, C. Yu, Z. Zhang, D. Yao, T. Sun, G. Jin, X. Cao, G. Cong, C. S. Jensen, X. Cheng, Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis, IEEE Transactions on Knowledge and Data Engineering 37 (1) (2025) 291–305. doi:10.1109/tkde.2024.3484454.
URL http://dx.doi.org/10.1109/tkde.2024.3484454

[5] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, in: Proceedings of the Twenty-

Seventh International Joint Conference on Artificial Intelligence, IJCAI-2018, International Joint Conferences on Artificial Intelligence Organization, 2018, p. 3634–3640. doi:10.24963/ijcai.2018/505.
URL http://dx.doi.org/10.24963/ijcai.2018/505

[6] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial-temporal graph modeling, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-2019, International Joint Conferences on Artificial Intelligence Organization, 2019, p. 1907–1913. doi:10.24963/ijcai.2019/264.
URL http://dx.doi.org/10.24963/ijcai.2019/264

[7] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018.
URL https://openreview.net/forum?id=SJiHXGWAZ

[8] C. Song, Y. Lin, S. Guo, H. Wan, Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting, Proceedings of the AAAI Conference on Artificial Intelligence 34 (01) (2020) 914–921. doi:10.1609/aaai.v34i01.5438.
URL http://dx.doi.org/10.1609/aaai.v34i01.5438

[9] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang, Connecting the dots: Multivariate time series forecasting with graph neural networks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery &; Data Mining, KDD '20, ACM, 2020, p. 753–763. doi:10.1145/3394486.3403118.
URL http://dx.doi.org/10.1145/3394486.3403118

[10] C. Shang, J. Chen, J. Bi, Discrete graph structure learning for forecasting multiple time series, in: 9th International Conference on Learning Representations, ICLR

2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net, 2021.
URL https://openreview.net/forum?id=WEHSlH5mOk

[11] F. Li, J. Feng, H. Yan, G. Jin, F. Yang, F. Sun, D. Jin, Y. Li, Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution, ACM Transactions on Knowledge Discovery from Data 17 (1) (2023) 1–21. doi:10.1145/3532611.
URL http://dx.doi.org/10.1145/3532611

[12] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong, Q. Zhang, Spectral temporal graph neural network for multivariate time-series forecasting, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020.
URL https://dl.acm.org/doi/10.5555/3495724.3497215

[13] Z. Shao, Z. Zhang, F. Wang, W. Wei, Y. Xu, Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting, in: Proceedings of the 31st ACM International Conference on Information &; Knowledge Management, CIKM '22, ACM, 2022, p. 4454–4458. doi:10.1145/3511808.3557702.
URL http://dx.doi.org/10.1145/3511808.3557702

[14] Z. Shao, Z. Zhang, F. Wang, Y. Xu, Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22, ACM, 2022, p. 1567–1577. doi:10.1145/3534678.3539396.
URL http://dx.doi.org/10.1145/3534678.3539396

[15] Y. Fang, Y. Qin, H. Luo, F. Zhao, B. Xu, L. Zeng, C. Wang, When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks, in: 2023 IEEE 39th International Conference on Data Engineering (ICDE), IEEE, 2023, p. 517–529. doi:10.1109/icde55515.2023.

00046.
URL http://dx.doi.org/10.1109/icde55515.2023.00046

[16] H. Liu, Z. Dong, R. Jiang, J. Deng, J. Deng, Q. Chen, X. Song, Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23, ACM, 2023, p. 4125–4129. doi:10.1145/3583780.3615160.
URL http://dx.doi.org/10.1145/3583780.3615160

[17] Z. Geng, J. Xu, R. Wu, C. Zhao, J. Wang, Y. Li, C. Zhang, Stgaformer: Spatial–temporal gated attention transformer based graph neural network for traffic flow forecasting, Information Fusion 105 (2024) 102228. doi:10.1016/j.inffus.2024.102228.
URL http://dx.doi.org/10.1016/j.inffus.2024.102228

[18] J. Deng, X. Chen, R. Jiang, X. Song, I. W. Tsang, St-norm: Spatial and temporal normalization for multi-variate time series forecasting, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 269–278. doi:10.1145/3447548.3467330.
URL https://doi.org/10.1145/3447548.3467330

[19] H. Gao, R. Jiang, Z. Dong, J. Deng, Y. Ma, X. Song, Spatial-temporal-decoupled masked pre-training for spatiotemporal forecasting, in: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI '24, 2024. doi:10.24963/ijcai.2024/442.
URL https://doi.org/10.24963/ijcai.2024/442

[20] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, in: Proceedings of the 34th Interna-

tional Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[21] J. Johnson, M. Douze, H. Jegou, Billion-scale similarity search with gpus, IEEE Transactions on Big Data 7 (3) (2021) 535–547. doi:10.1109/tbdata.2019.2921572.
URL http://dx.doi.org/10.1109/tbdata.2019.2921572

[22] Y. N. Dauphin, A. Fan, M. Auli, D. Grangier, Language modeling with gated convolutional networks, in: Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, JMLR.org, 2017, p. 933–941.
URL https://dl.acm.org/doi/10.5555/3305381.3305478

[23] Z. Dong, R. Jiang, H. Gao, H. Liu, J. Deng, Q. Wen, X. Song, Heterogeneity-informed meta-parameter learning for spatiotemporal time series forecasting, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '24, ACM, 2024, p. 631–641. doi:10.1145/3637528.3671961.
URL http://dx.doi.org/10.1145/3637528.3671961

[24] J. Hu, T. Zheng, L. Peng, F. Teng, S. Du, T. Li, Lightst: A simplifying spatio-temporal graph neural network for traffic flow forecasting, IEEE Transactions on Big Data 11 (5) (2025) 2517–2528. doi:10.1109/tbdata.2025.3544131.
URL http://dx.doi.org/10.1109/tbdata.2025.3544131

[25] J. Jiang, C. Han, W. X. Zhao, J. Wang, Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction, Proceedings of the AAAI Conference on Artificial Intelligence 37 (4) (2023) 4365–4373. doi:10.1609/aaai.v37i4.25556.
URL http://dx.doi.org/10.1609/aaai.v37i4.25556

[26] Y. Hou, D. Zhang, Graph neural network-enhanced multivariate time series forecasting with series-core fusion, in: 2025 8th International Conference on Advanced

Algorithms and Control Engineering (ICAACE), IEEE, 2025, p. 1850–1854. doi:10.1109/icaace65325.2025.11019045.
URL http://dx.doi.org/10.1109/icaace65325.2025.11019045

[27] L. Zhong, B. Wang, Z. Tian, T. K. Rodrigues, W. Liu, W. She, Multiview spatiotemporal dynamic graph convolution network for traffic flow prediction, IEEE Internet of Things Journal 12 (18) (2025) 37062–37076. doi:10.1109/JIOT.2025.3580788.

[28] J. Chen, H. Ye, Z. Ying, Y. Sun, W. Xu, Dynamic trend fusion module for traffic flow prediction, Applied Soft Computing 174 (2025) 112979. doi:10.1016/j.asoc.2025.112979.
URL http://dx.doi.org/10.1016/j.asoc.2025.112979

[29] Y. Liu, C. Li, J. Wang, M. Long, Koopa: Learning non-stationary time series dynamics with koopman predictors, in: Advances in Neural Information Processing Systems 36, NeurIPS 2023, Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023, p. 12271–12290. doi:10.52202/075280-0538.
URL http://dx.doi.org/10.52202/075280-0538