# A Retrieval Augmented Spatio-Temporal Framework for Traffic Prediction

**Weilin Ruan**[1], **Xilin Dang**[2], **Ziyu Zhou**[1], **Sisuo Lyu**[1], **Yuxuan Liang**[1*]
[1] The Hong Kong University of Science and Technology (Guangzhou)
[2] The Chinese University of Hong Kong
{rwlinno,ziyuzhou30}@gmail.com, xldang23@cse.cuhk.edu.hk, {sisuolyu,yuxliang}@outlook.com

## Abstract

Traffic prediction is a cornerstone of modern intelligent transportation systems and a critical task in spatio-temporal forecasting. Although advanced Spatio-temporal Graph Neural Networks (STGNNs) and pre-trained models have achieved significant progress in traffic prediction, two key challenges remain: (i) limited contextual capacity when modeling complex spatio-temporal dependencies, and (ii) low predictability at fine-grained spatio-temporal points due to heterogeneous patterns. Inspired by Retrieval-Augmented Generation (RAG), we propose RAST, a universal framework that integrates retrieval-augmented mechanisms with spatio-temporal modeling to address these challenges. Our framework consists of three key designs: 1) Decoupled Encoder and Query Generator to capture decoupled spatial and temporal features and construct a fusion query via residual fusion; 2) Spatio-temporal Retrieval Store and Retrievers to maintain and retrieve vectorized fine-grained patterns; and 3) Universal Backbone Predictor that flexibly accommodates pre-trained STGNNs or simple MLP predictors. Extensive experiments on six real-world traffic networks, including large-scale datasets, demonstrate that RAST achieves superior performance while maintaining computational efficiency.

## 1 Introduction

Traffic prediction stands as a cornerstone of modern Intelligent Transportation Systems (ITS), enabling critical applications including traffic management, route optimization, and congestion mitigation (Chavhan and Venkataram 2020; Zheng et al. 2014). The accurate forecasting of traffic conditions directly impacts urban mobility, economic efficiency, and environmental sustainability across metropolitan areas worldwide. Spatio-temporal Forecasting (STF) provides the methodological foundation for addressing these traffic prediction challenges, as traffic data inherently exhibits complex interdependencies across both spatial and temporal dimensions (Wang, Cao, and Philip 2020; Jin et al. 2023).

The evolution of STF methodologies has progressed from traditional statistical approaches (Box et al. 2015; Lütkepohl 2005; Chandra and Al-Deek 2009) to sophisticated deep learning architectures (Shi et al. 2015; Kipf and Welling 2016; Hochreiter and Schmidhuber 1997) as application scenarios have become increasingly complex. This progression culminated in the development of Spatio-temporal Graph Neural Networks (STGNNs) (Wu et al. 2020; Yu, Yin, and Zhu 2017; Bai et al. 2020), which have achieved remarkable success in modeling complex spatial-temporal dependencies by representing traffic networks as graphs and leveraging graph convolution operations (Li et al. 2017; Wu et al. 2019). More recently, the emergence of Large Models (LMs) and pre-trained models from Natural Language Processing (NLP) (Devlin 2018; Jin et al. 2021) and Computer Vision (CV) (Dosovitskiy 2020; He et al. 2022) has opened new opportunities for enhancing spatio-temporal forecasting capabilities (Zhou et al. 2024; Jin et al. 2024b; Yan et al. 2024).

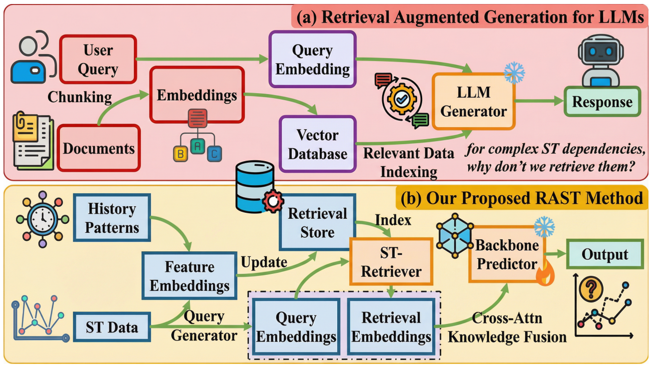

Figure 1: Motivation of RAST. Inspired by retrieval-augmented generation for large language models, we design a specialized retrieval-augmented framework for STF tasks.

Despite these advances, current STF approaches face two critical limitations: **(i) Limited Contextual Capacity vs. scale of ST data:** Contemporary pre-trained STGNNs suffer from constrained contextual embedding capacity when handling complex spatio-temporal dependencies in large-scale traffic networks (Jin et al. 2024a; Jiang 2023; Liu et al. 2023b). Drawing inspiration from retrieval-augmented generation (RAG) that has shown promise in addressing context limitations in Large Language Models (LLMs) (Lewis et al. 2020), we investigate whether retrieval-augmented mechanisms can compensate for the limited spatio-temporal learning capacity, as illustrated in Figure 1; **(ii) Complex Ar-**

---

[*] Corresponding author: Yuxuan Liang.

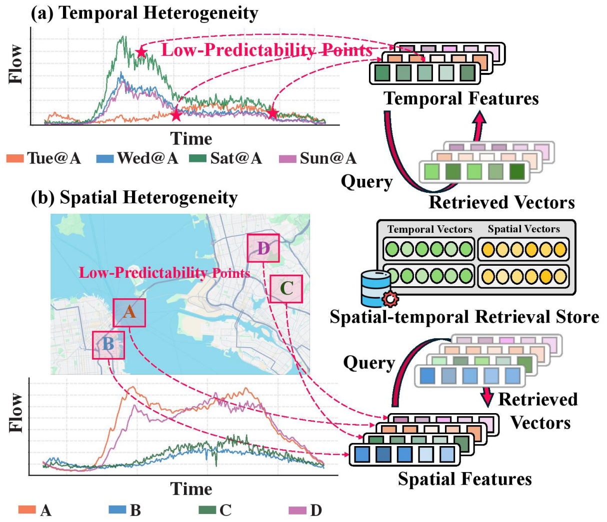

Figure 2: The spatial and temporal heterogeneity problem (a and b). Our methods utilize a retrieval-augmented mechanism with dual-dimension vector storage. By extracting low-predictable data points and retrieving history patterns, we improve model capability in those challenging cases.

**chitecture vs. Low Predictability:** Due to the inherent heterogeneity in spatio-temporal data (Jin et al. 2023), existing STF approaches lack efficient mechanisms for fine-grained pattern adjustment within limited embedding lengths (Jin et al. 2023, 2024a; Jiang 2023). Current performance improvements of the STGNNs rely heavily on complex model architectures (Shao et al. 2022c; Lan et al. 2022) to capture overall trends, yet low-predictability points in both temporal and spatial dimensions remain difficult to capture. Instead of further increasing model complexity by adding more weighted parameters, we propose to capture complex spatio-temporal dependencies through explicit memory storage and retrieval mechanisms, as demonstrated in Figure 2.

To bridge the gap in applying RAG to STF and further address the aforementioned challenges, we propose RAST (Retrieval-Augmented Spatio-Temporal forecasting), a universal framework that integrates retrieval-augmented mechanisms with spatio-temporal modeling for traffic prediction and pre-trained model enhancements. Specifically, ❶ our approach maintains a vector-based dual-dimension spatio-temporal retrieval store. During training, our method decouples input data into spatial and temporal encodings used to update the store while constructing the context-aware queries through residual fusion. ❷ Specific retrievers then search and project dual-dimension retrieval embeddings under queries. ❸ Finally, we fuse decoupled retrieval embeddings with current queries through a cross-attention module and obtain final predictions via the universal backbone predictor. Our key contributions are summarized as follows:

- **A Universal Retrieval-Augmented Framework for Spatio-temporal Forecasting:** We introduce RAST, the first retrieval-augmented framework specifically designed for spatio-temporal forecasting while providing a universal framework for existing pre-trained STGNNs as an enhancement method without expanding the model capacity.
- **A Spatio-temporal Retrieval Store and ST-Retriever for Low-Predictability Patterns:** We design a spatio-temporal retrieval store that vectorizes dual-dimension features and maintains them within memory banks, combined with optimization techniques for efficient memory and retrieval of complex spatio-temporal patterns.
- **Comprehensive Empirical Validation:** Extensive experiments on six real-world datasets demonstrate that our proposed method effectively captures complex spatio-temporal patterns while maintaining high computational efficiency, achieving up to 24.75% improvement in average MAE compared to the RPMixer on the SD dataset.

## 2 Related Work

**Spatio-temporal Forecasting (STF)** is a fundamental task in numerous application domains such as traffic management, urban planning, and environmental monitoring (Lv et al. 2014; Jin et al. 2023; Bi et al. 2023; Jiang et al. 2021; Zhou et al. 2025). Early deep learning methods combined CNNs and RNNs (Shi et al. 2015; Hochreiter and Schmidhuber 1997) but struggled with non-Euclidean traffic networks. The emergence of Spatio-temporal Graph Neural Networks (STGNNs) addressed this limitation by integrating Graph Neural Networks (GNNs) (Kipf and Welling 2016) with temporal models (Li et al. 2017; Zhang, Zheng, and Qi 2017; Yu, Yin, and Zhu 2017; Bai et al. 2020; Guo et al. 2019; Shao et al. 2022c; Liang et al. 2018; Wu et al. 2020, 2019; Zheng et al. 2020). Despite these advancements, the performance improvements of STGNNs have begun to plateau due to limited contextual capacity and reliance on increasingly complex architectures (Wang et al. 2020; Jin et al. 2024a). This stagnation has prompted research toward integrating pre-trained models and Large Language Models (LLMs) (Zhou et al. 2024; Liu et al. 2023a; Jin et al. 2024b; Yuan et al. 2024; Shao et al. 2022b) to enhance predictive capabilities. *However, these methods still struggle to capture spatio-temporal heterogeneity in large-scale scenarios, while universal solutions for contextual capacity limitations in spatio-temporal modeling remain largely unexplored.*

**Retrieval-Augmented Generation (RAG)** has emerged as a transformative paradigm for enhancing large language model performance, particularly in knowledge-intensive tasks (Kandpal et al. 2023; Lewis et al. 2020; Gao et al. 2023). Traditional LLMs struggle with tasks requiring vast amounts of factual knowledge or domain-specific expertise due to finite parametric memory and limited context (Roberts, Raffel, and Shazeer 2020; Petroni et al. 2019). RAG addresses this limitation by integrating external retrieval mechanisms that enable models to dynamically access relevant information from large knowledge bases during inference (Zhu et al. 2024; Izacard and Grave 2021; Borgeaud et al. 2022), improving open-domain question answering, fact-checking, and few-shot learning tasks (Ram et al. 2023; Shi et al. 2023). *While the retrieval-augmented mechanism has been extensively studied in natural language*

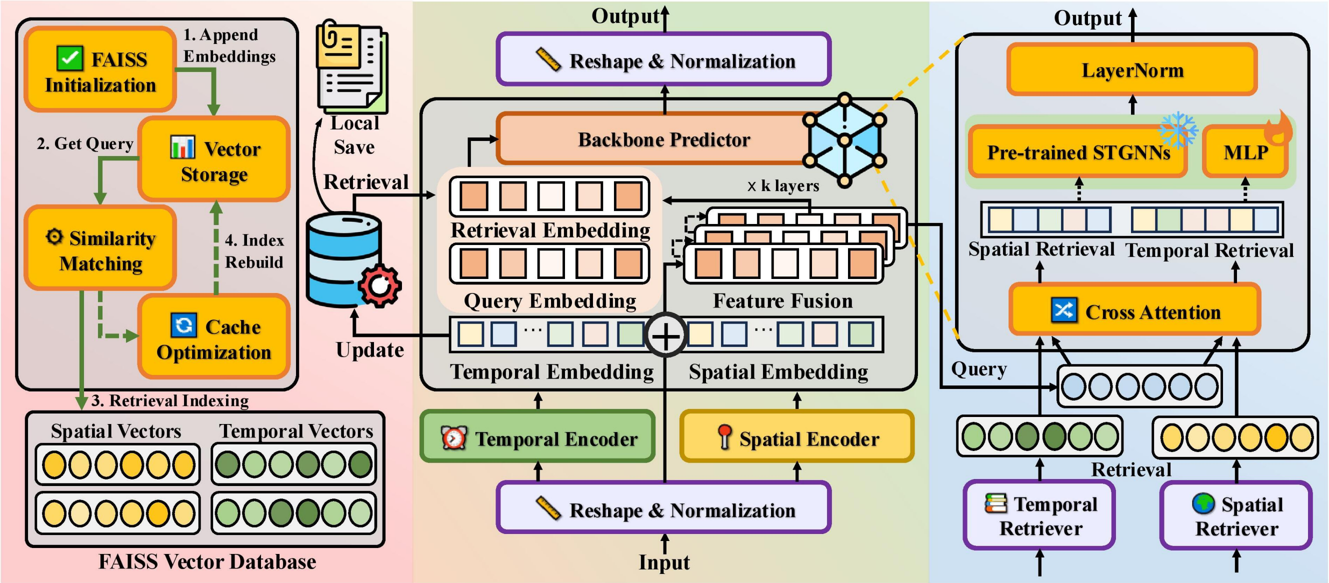

Figure 3: Overview of the RAST framework. RAST integrates retrieval mechanisms with spatio-temporal modeling to enhance performance by maintaining and utilizing historical patterns. The proposed approach addresses the fundamental challenge of capturing long-term dependencies and complex spatio-temporal correlations in time series data.

*processing, its application to vast spatio-scenarios presents significant opportunities, particularly given the similar challenges of constrained model capacity.*

## 3 Methodology

In this section, we present **RAST**, a Retrieval Augmented Spatio-Temporal forecasting framework as illustrated in Figure 3. Our approach consists of five core components working synergistically: *1) Decoupled Encoder Layers* that convert raw inputs into vectorized embedding representations, *2) Query Generator* that constructs fusion queries via residual fusion, *3) Retrieval Store* that maintains and fine-grained historical patterns *4) ST-Retriever* that retrieves embeddings based on queries and further fuses them with cross-attention blocks, and *5) Backbone Predictor* that accommodates diverse pre-trained STGNNs or simple predictors (e.g. MLP) for enhanced forecasting performance.

### 3.1 Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ denote a spatio-temporal graph where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ represents the set of $N$ spatial nodes. $\mathcal{E}$ represents the edges connecting spatially related nodes, and $\mathcal{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix encoding spatial relationships. At each time step $t$, we observe a $D_{in}$ feature dimension matrix $X_t \in \mathbb{R}^{N \times D_{in}}$ at time $t$. Given historical observations $\mathbf{X} = \{X_{t-L+1}, X_{t-L+2}, \ldots, X_t\} \in \mathbb{R}^{L \times N \times D_{in}}$ over $L$ time steps, the spatio-temporal forecasting task aims to predict future observations $\mathbf{Y} \in \mathbb{R}^{H \times N \times D_{out}}$ over the next $H$ horizons. The external memory bank $\mathcal{M}$ is introduced and dynamically updated in our approach. Formally, we seek to learn a mapping function with the retrieval-augmented mechanism:

$$f_\theta : \mathbf{X} \times \mathcal{G} \times \mathcal{M} \rightarrow \mathcal{Y} \tag{1}$$

where $\theta$ represents the learnable parameters of the model.

### 3.2 Data Encoding and Query Construction

**Dual-dimension Feature Disentanglement.** Following recent advances in spatio-temporal modeling (Shao et al. 2022a), we employ decoupled encoder layers to separately process temporal and spatial information. This encoding module initially captures basic characteristics (e.g., cyclicity for temporal, regionality for spatial) formulated as follows:

$$\mathbf{E}_{tp} = \sigma(\text{Conv2D}(\mathbf{X})) \in \mathbb{R}^{B \times N \times D_{tp}} \tag{2}$$

$$\mathbf{E}_{sp} = \sigma(\mathbf{W}_{sp}(\mathbf{X}, \mathcal{G})) \in \mathbb{R}^{B \times N \times D_{sp}} \tag{3}$$

where $\sigma(\cdot)$ denotes the reshape and normalization operation for the corresponding dimension. $D_{tp}, D_{sp}$ are temporal and spatial feature dimension. The 2D convolutional kernel is initialized using Kaiming normal initialization to ensure stable training dynamics, while the spatial transformation matrix $\mathbf{W}_{sp}$ is initialized with Xavier uniform distribution.

**Context-Aware Query Generation.** To construct an input that retrieves the most similar embedding, we designed a specific query generator. Specifically, the temporal and spatial embeddings are concatenated and projected to a fusion representation $\mathbf{E}_f^{(0)}$. Then we construct context-aware query representations $\mathcal{Q}_{st}$ through $L$ encoder layers with residual connections as follows:

$$\mathbf{E}_f^{(0)} = \text{Linear}_Q(\text{Concat}[\mathbf{E}_{sp}; \mathbf{E}_{tp}]), \tag{4}$$

$$\mathbf{E}_f^{(l+1)} = \text{LayerNorm}(\mathbf{E}_f^{(l)} + \text{FFN}(\mathbf{E}_f^{(l)})), \tag{5}$$

$$\mathcal{Q}_{st} = \mathbf{E}_f^{(L)} \in \mathbb{R}^{B \times N \times D_q} \tag{6}$$

where FFN denotes a feed-forward network with ReLU activation and dropout for regularization.

### 3.3 Spatio-temporal Retrieval Store

**Pattern Indexing and Storage.** Traditional spatio-temporal models suffer from limited contextual capacity when handling complex dependencies, while we introduce a dual-dimension memory bank $\mathcal{M} = \{\mathcal{M}_{sp}, \mathcal{M}_{tp}\}$ that dynamically maintains vectorized historical patterns:

$$\mathcal{M}_{sp}^{(i)} = \{\mathbf{v}_{sp}^{(i)}, \mathbf{m}_{sp}^{(i)}\}_{i=1}^{|\mathcal{M}_{sp}|} \tag{7}$$

$$\mathcal{M}_{tp}^{(j)} = \{\mathbf{v}_{tp}^{(j)}, \mathbf{m}_{tp}^{(j)}\}_{j=1}^{|\mathcal{M}_{tp}|} \tag{8}$$

where $\mathbf{v}^{(i)}, \mathbf{v}^{(j)}$ represent chunked embedding vectors and $\mathbf{m}^{(i)}, \mathbf{m}^{(j)}$ contain associated metadata including statistical summaries and importance measures for sustainable storage.

To enable fast similarity search, our retrieval store utilizes the Facebook AI Similarity Search (FAISS) library (Douze et al. 2024; Johnson, Douze, and Jégou 2019) for efficient similarity-based indexing. Given history queries $\mathcal{Q}_{st}$ and current state $e$, we maintain and compute indices $\mathcal{I}$ for temporal and spatial embeddings (more details in Appendix D):

$$\mathcal{I}_{sp} = \sigma(\text{Index}(\{v_{sp}^{(e)}\} \in \mathcal{M}_{sp}|\mathcal{Q}_{st})) \tag{9}$$

$$\mathcal{I}_{tp} = \sigma(\text{Index}(\{v_{tp}^{(e)}\} \in \mathcal{M}_{tp}|\mathcal{Q}_{st})) \tag{10}$$

where $v_s^{(e)}, v_t^{(e)}$ represent the sampled decoupled vectors and $\sigma(\cdot)$ denotes the operation of discretization. The indices support (i) periodic rebuild, (ii) LRU caching, and (iii) GPU Acceleration, significantly improving retrieval efficiency.

**Information-Theoretic ST-Retriever.** To identify and select the most relevant historical patterns of vectors from the retrieval store, we defined spatio-temporal retrievers to search for the Top-k most relevant information based on the similarity searching function Retriever$(\cdot)$. Given a context-aware query $\mathcal{Q} \in \mathbb{R}^{B\times N\times D_q}$ and the computed indices $\mathcal{I}$, the retriever performs fine-grained pattern discovery utilizing L2 distance as follows:

$$\mathcal{D}(\mathcal{Q}, \mathbf{v}_i) = -||\mathcal{Q} - \mathbf{v}_i||_2^2 \tag{11}$$

$$\text{Retriever}(\mathcal{Q}, \mathcal{I}, k) = \arg\max_k \{\mathcal{D}(\mathcal{Q}, \mathbf{v}_j)\}_{j=1}^{|\mathbf{V}|} \tag{12}$$

where $\mathbf{v}$ represents chunked pattern vectors in the memory bank $\mathcal{M}$ indexing by$\mathcal{M}(\mathcal{I}) \mapsto \mathbf{V}$.

Given dual-dimension feature $\mathbf{E}_{sp}, \mathbf{E}_{tp}$ encoded before, the set of indices $\mathcal{I}$ of lengths $k$, the ST-retrievers match the fine-grained pattern sets in the retrieval store and calculate the momentum of the memory banks as follows:

$$\mathcal{E}_s = \text{Retriever}(\mathbf{E}_{sp}, \mathcal{I}_s, k) = \{(\mathbf{v}_s^{(i)}, \omega_s^{(i)})\}_{i=1}^{k} \tag{13}$$

$$\mathcal{E}_t = \text{Retriever}(\mathbf{E}_{tp}, \mathcal{I}_t, k) = \{(\mathbf{v}_t^{(j)}, \omega_t^{(j)})\}_{j=1}^{k} \tag{14}$$

To enhance the quality of retrieved vectors $\mathbf{v}_i \in \{\mathcal{E}_s, \mathcal{E}_t\}$, we incorporate the similarity score $\mathbf{s}_i = \mathcal{D}(\mathcal{Q}, \mathbf{v}_i)$ and momentum scores $\omega_i$ that enable weighted pattern aggregation. Given information entropy function $\mathcal{H}(\mathbf{v}) = -\sum_{d=1}^{D} p_d \log p_d$ where $p_d = \frac{\exp(\mathbf{v}_d)}{\sum_{j=1}^{D}\exp(\mathbf{v}_j)}$ measures the information entropy, the momentum scores are updated with the diversity-similarity coefficient $\lambda$ and the temperature parameter $\tau$ for confidence calibration as follows:

$$\omega_i' = \omega_i + \text{softmax}\left(s_i + \lambda \cdot \mathcal{H}(\mathbf{v_i})/\tau\right) \tag{15}$$

**Momentum-Based Memory Management.** The retrieval store is updated periodically during training with a defined interval to balance between pattern freshness and computational overhead. To prevent unbounded memory growth while preserving both recent and historically significant patterns, we implement adaptive memory management:

$$\mathcal{M}_s^{(e+1)} = (1 - \omega^s)\mathcal{M}_s^{(e)} + \omega_s \cdot \sigma(\mathcal{E}_s) \tag{16}$$

$$\mathcal{M}_t^{(e+1)} = (1 - \omega^t)\mathcal{M}_t^{(e)} + \omega_t \cdot \sigma(\mathcal{E}_t) \tag{17}$$

where $\sigma(\cdot)$ denotes an insertion, and adaptive memory parameters $\alpha, \beta$ are determined by similarity scores $\mathbf{s}$, ensuring optimal balance between memory freshness and stability.

**Cross-Attention Knowledge Fusion.** Rather than simply taking the retrieval vectors separated by temporal and spatial dimensions, we employ multi-head attention mechanisms to further fuse query embeddings with retrieved patterns:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\Big(\frac{\mathbf{Q}\,\mathbf{K}^\top}{\sqrt{d_k}}\Big)\,\mathbf{V} \tag{18}$$

$$\mathcal{R}_t = \text{Attn}(\mathcal{Q}_{st}, \mathcal{E}_t, \mathcal{E}_t), \quad \mathcal{R}_s = \text{Attn}(\mathcal{Q}_{st}, \mathcal{E}_s, \mathcal{E}_s) \tag{19}$$

$$\mathcal{R}_f = \text{Attn}(\mathcal{Q}_{st}, \mathcal{R}_s, \mathcal{R}_t), \quad \mathbf{H}_f = \text{Concat}[\mathcal{Q}_{st}; \mathcal{R}_f] \tag{20}$$

where $\mathcal{R}_f \in \mathbb{R}^{B\times N\times D_r}$ denotes the fused retrieval with embedding dimension $D_r$ and $\mathbf{H}_f$ is the fusion embedding that preserves the original query while combining with the most relevant dual-dimension retrieved patterns.

### 3.4 Prediction and Optimization

**Universal Backbone Predictor.** For prediction generation, we utilized function $\mathcal{B}(\mathbf{X}, \mathbf{H}_f, \mathcal{G}) \mapsto \mathcal{Y}$, leveraging a universal backbone network $\mathcal{B}(\cdot)$ for fine-tuning tasks or training from scratch. We designed a universal interface that allows frozen or learnable pre-trained STGNNs to use our retrieval-augmented mechanisms, accommodating various backbone configurations without modification to the core retrieval mechanism for improvements. And for the following experiments, we default to applying the lightweight Multilayer Perceptron (MLP) as $\mathcal{B}$ in the fair perspective.

**Prediction Generation.** We employ a residual enhancement pipeline preserving information flow while enabling architectural flexibility, with layer normalization and feed-forward operation after. The combined feature representation $\mathbf{Z} \in \mathbb{R}^{B\times N\times(D_q+D_r)}$ is processed through the backbone predictor to generate final predictions:

$$\mathbf{Z} = \mathcal{B}(\mathbf{H}_f)||\text{Conv}(\sigma(\text{Conv}(\mathbf{H}_f \cdot W_1 + b_1))W_2 + b_2) \tag{21}$$

$$\hat{\mathcal{Y}} = \text{LayerNorm}(\mathbf{Z}) + \text{FFN}(\mathbf{Z}) \tag{22}$$

**Loss Function.** The model is trained using Mean Absolute Error (MAE) loss with L2 regularization:

$$\mathcal{L} = \frac{1}{M}\sum_{i\in M}||\hat{\mathcal{Y}}_i - \mathcal{Y}_i|| + \lambda||\theta||^2 \tag{23}$$

where $\lambda$ is the regularization coefficient for model parameter $\theta$, and $M$ denotes the collection of valid data points.

| Methods | PEMS03 | | | PEMS04 | | | PEMS07 | | | PEMS08 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MAE | RMSE | MAPE(%) | MAE | RMSE | MAPE(%) | MAE | RMSE | MAPE(%) | MAE | RMSE | MAPE(%) |
| ARIMA | 35.31 | 47.59 | 33.78 | 33.73 | 48.80 | 24.18 | 38.17 | 59.27 | 19.46 | 31.09 | 44.32 | 22.73 |
| VAR | 23.65 | 38.26 | 24.51 | 23.75 | 36.66 | 18.09 | 75.63 | 115.24 | 32.22 | 23.46 | 36.33 | 15.42 |
| SVR | 21.97 | 35.29 | 21.51 | 28.70 | 44.56 | 19.20 | 32.49 | 50.22 | 14.26 | 23.25 | 36.16 | 14.64 |
| LSTM | 21.33 | 35.11 | 23.33 | 27.14 | 41.59 | 18.20 | 29.98 | 45.84 | 13.20 | 22.20 | 34.06 | 14.20 |
| TCN | 19.31 | 33.24 | 19.86 | 31.11 | 37.25 | 15.48 | 32.68 | 42.23 | 14.22 | 22.69 | 35.79 | 14.04 |
| Transformer | 17.50 | 30.24 | 16.80 | 23.83 | 37.19 | 15.57 | 26.80 | 42.95 | 12.11 | 18.52 | 28.68 | 13.66 |
| DCRNN | 18.18 | 30.31 | 18.91 | 24.70 | 38.12 | 17.12 | 25.30 | 38.58 | 11.66 | 17.86 | 27.83 | 11.45 |
| STGCN | 17.49 | 30.12 | 17.15 | 22.70 | 35.55 | 14.59 | 25.38 | 38.78 | 11.08 | 18.02 | 27.83 | 11.40 |
| ASTGCN | 17.69 | 29.66 | 19.40 | 22.93 | 35.22 | 16.56 | 28.05 | 42.57 | 13.92 | 18.61 | 28.16 | 13.08 |
| GWNet | 19.85 | 32.94 | 19.31 | 25.45 | 39.70 | 17.29 | 26.85 | 42.78 | 12.12 | 19.13 | 31.05 | 12.68 |
| LSGCN | 17.94 | 29.85 | 16.98 | 21.53 | 33.86 | 13.18 | 27.31 | 41.16 | 11.98 | 17.73 | 26.76 | 11.30 |
| STSGCN | 17.48 | 29.21 | 16.78 | 21.19 | 33.65 | 13.90 | 24.26 | 39.03 | 10.21 | 17.13 | 26.80 | 10.96 |
| STFGNN | 16.77 | 28.34 | 16.30 | 19.83 | 31.88 | 13.02 | 22.07 | 35.80 | 9.21 | 16.64 | 26.22 | 10.60 |
| STGODE | 16.50 | 27.84 | 16.69 | 20.84 | 32.82 | 13.77 | 22.99 | 37.54 | 10.14 | 16.81 | 25.97 | 10.62 |
| DSTAGNN | 15.57 | 27.21 | **14.68** | 19.30 | 31.46 | 12.70 | 21.42 | 34.51 | 9.01 | 15.67 | 24.77 | 9.94 |
| EnhanceNet | 16.05 | 28.33 | 15.83 | 20.44 | 32.37 | 13.58 | 21.87 | 35.57 | 9.13 | 16.33 | 25.46 | 10.39 |
| AGCRN | 16.06 | 28.49 | 15.85 | 19.83 | 32.26 | 12.97 | 21.29 | 35.12 | 8.97 | 15.95 | 25.22 | 10.09 |
| Z-GCNETs | 16.64 | 28.15 | 16.39 | 19.50 | 31.61 | 12.78 | 21.77 | 35.17 | 9.25 | 15.76 | 25.11 | 10.01 |
| NHiTS | 20.57 | 35.01 | 20.28 | 27.54 | 42.95 | 18.88 | 29.08 | 44.87 | 12.64 | 21.75 | 33.97 | 13.61 |
| TimeMixer | 20.95 | 32.64 | 27.71 | 27.37 | 40.60 | 27.26 | 30.52 | 44.86 | 19.45 | 21.90 | 33.61 | 17.56 |
| TAMP | 16.46 | 28.44 | 15.37 | 19.74 | 31.74 | 13.22 | 21.84 | 35.42 | 9.24 | 16.36 | 25.98 | 10.15 |
| iTransformer | 17.31 | 27.79 | 16.53 | 23.18 | 38.02 | 15.32 | 23.66 | 39.85 | 9.90 | 16.28 | 27.84 | 10.53 |
| STKD | 16.03 | 25.95 | 15.76 | 19.86 | 31.93 | 13.18 | 21.64 | 34.96 | 9.03 | 15.81 | 25.07 | 10.02 |
| STDN | 17.77 | 28.63 | 21.37 | 20.86 | 32.63 | 15.26 | 20.08 | 33.73 | 9.29 | 19.19 | 28.53 | 14.99 |
| **RAST (Ours)** | **15.20** | **25.81** | 16.12 | **18.39** | **29.93** | **12.43** | **19.52** | **32.73** | **8.23** | **14.16** | **23.33** | **9.27** |

Table 1: Performance comparison assessed by averaging over all 12 prediction steps with baseline models on the PEMS03, 04, 07, 08 datasets. **Bold**: best; Underline: second best.

## 4 Experiments

We conduct extensive experiments to evaluate the effectiveness of RAST in multiple ways. These experiments are designed to answer the following Research Questions (RQ):

- **RQ1**: How does RAST perform compared to state-of-the-art spatio-temporal forecasting methods?
- **RQ2**: What is the individual contribution of each component in the RAST framework?
- **RQ3**: How sensitive is RAST to key hyperparameters?
- **RQ4**: What is the computational efficiency of RAST compared to baseline methods?

### 4.1 Experimental Settings

**Datasets and Baselines.** We conducted comprehensive experiments across six diverse traffic networks: 1) PEMS03, PEMS04, PEMS07, PEMS08 (Song et al. 2020) datasets, and 2) the large-scale dataset San Diego (SD), Greater Bay Area (GBA) as introduced in LargeST (Liu et al. 2023b). Detailed statistics of these datasets are given in Appendix B. We compare our RAST with 21 classic or advanced baselines, which are categorized into 3 groups: **(i) Non-spatial methods:** ARIMA (Box et al. 2015)VAR (Lütkepohl 2005), SVR (Awad and Khanna 2015), LSTM (Hochreiter and Schmidhuber 1997), TCN (Lea et al. 2017), Transformer (Vaswani et al. 2017), NHiTS (Challu et al. 2022), iTransformer (Liu et al. 2023c), TimeMixer (Wang et al. 2024a); **(ii) GNN-based Spatial-temporal Models** DCRNN (Li et al. 2017), STGCN (Yu, Yin, and Zhu 2017), ASTGCN (Guo et al. 2019), GWNet (Wu et al. 2019), LSGCN (Huang et al. 2020), STSGCN (Song et al. 2020), STFGNN (Li and Zhu 2021), STGODE (Fang et al. 2021), DSTAGNN (Lan et al. 2022), AGCRN (Bai et al. 2020), D2STGNN (Shao et al. 2022c), and **(iii) Other Enhanced Approaches:** Z-GCNETs (Chen et al. 2021), TAMP (Chen et al. 2022), STKD (Wang et al. 2024b), RPMixer (Yeh et al. 2024), STDN (Cao et al. 2025). More details of the baselines are given in Appendix A.

**Metrics and Settings.** Performance is evaluated using standard metrics, including MAE, RMSE, and MAPE. We use 12 historical time steps to forecast the next 12 steps and calculate the average across horizons 3, 6, and 12. For the fairness of the experiment, we only utilized the simple MLP trained from scratch as the backbone predictor in the following experiments. All models are implemented in PyTorch and trained on Nvidia RTX A6000 GPUs with consistent hyperparameter tuning protocols to ensure fair comparison. The models are trained using the Adam optimizer with a learning rate of 0.002, a batch size of 32, and a maximum of 300 epochs, applying an early stopping strategy. More detailed experimental settings are provided in Appendix C.

### 4.2 Performance Evaluation (RQ1)

Table 1 presents a comprehensive comparison of various approaches for traffic forecasting tasks across PEMS datasets. While the results on PEMS03 show room for improvement, RAST consistently outperforms state-of-the-art baseline methods on the remaining datasets. On the

| Dataset | Methods | Horizon 3 | | | Horizon 6 | | | Horizon 12 | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAPE (%) | MAE | RMSE | MAPE (%) | MAE | RMSE | MAPE (%) | MAE | RMSE | MAPE (%) |
| SD | LSTM | 19.03 | 30.53 | 11.81 | 25.84 | 40.87 | 16.44 | 37.63 | 59.07 | 25.45 | 26.44 | 41.73 | 17.20 |
| | DCRNN | 17.14 | 27.47 | 11.12 | 20.99 | 33.29 | 13.95 | 26.99 | 42.86 | 18.67 | 21.03 | 33.37 | 14.13 |
| | STGCN | 17.45 | 29.99 | 12.42 | 19.55 | 33.69 | 13.68 | 23.21 | 41.23 | 16.32 | 19.67 | 34.14 | 13.86 |
| | ASTGCN | 19.56 | 31.33 | 12.18 | 24.13 | 37.95 | 15.38 | 30.96 | 49.17 | 21.98 | 23.70 | 37.63 | 15.65 |
| | STGODE | 16.75 | 28.04 | 11.00 | 19.71 | 33.56 | 13.16 | 23.67 | 42.12 | 16.58 | 19.55 | 33.57 | 13.22 |
| | DSTAGNN | 18.13 | 28.96 | 11.38 | 21.71 | 34.44 | 13.93 | 27.51 | 43.95 | 19.34 | 21.82 | 34.68 | 14.40 |
| | RPMixer | 18.54 | 30.33 | 11.81 | 24.55 | 40.04 | 16.51 | 35.90 | 58.31 | 27.67 | 25.25 | 42.56 | 17.64 |
| | **RAST (Ours)** | **15.84** | **26.41** | **10.15** | **18.55** | **31.56** | **12.13** | **22.18** | **39.43** | **15.38** | **18.39** | **31.96** | **12.19** |
| GBA | LSTM | 20.38 | 33.34 | 15.47 | 27.56 | 43.57 | 23.52 | 39.03 | 60.59 | 37.48 | 27.96 | 44.21 | 24.48 |
| | DCRNN | 18.71 | 30.36 | 14.72 | 23.06 | 36.16 | 20.45 | 29.85 | 46.06 | 29.93 | 23.13 | 36.35 | 20.84 |
| | STGCN | 21.05 | 34.51 | 16.42 | 23.63 | 38.92 | 18.35 | 26.87 | 44.45 | 21.92 | 23.42 | 38.57 | 18.46 |
| | ASTGCN | 21.46 | 33.86 | 17.24 | 26.96 | 41.38 | 24.22 | 34.29 | 52.44 | 32.53 | 26.47 | 40.99 | 23.65 |
| | DSTAGN | 19.73 | 31.39 | 15.42 | 24.21 | 37.70 | 20.99 | 30.12 | 46.40 | 28.16 | 23.82 | 37.29 | 20.16 |
| | RPMixer | 20.31 | 33.34 | 15.64 | 26.95 | 44.02 | 22.75 | 39.66 | 66.44 | 37.35 | 27.77 | 47.72 | 23.87 |
| | **RAST (Ours)** | **17.71** | **29.29** | **13.72** | **20.86** | **34.26** | **16.63** | **24.97** | **41.33** | **21.25** | **20.64** | **34.47** | **16.63** |

Table 2: Large-scale traffic forecasting performance comparison of our RAST and baselines on the SD and GBA datasets. Our RAST achieves the best performance across all prediction horizons and metrics. **Bold**: best; Underline: second best.

PEMS07 dataset, RAST achieves a MAE of 19.52, surpassing DSTAGNN by 8.87%. On the PEMS08 dataset, RAST reduces RMSE by 1.65 compared to competitive STKD, while on the PEMS04, it outperforms all baselines with an MAE of 18.39 and RMSE of 29.93.

Table 2 extends our evaluation to larger datasets, while the performance on the horizon 3, horizon 6, horizon 12, and the average of the whole 12 horizons are reported. Our method maintains its advantage on larger traffic networks, achieving the best results at each horizon. On the SD dataset, RAST achieves an average MAE of 18.39, RMSE of 31.96, and MAPE of 12.19%, representing improvements compared to the second-best method STGODE. For the long-term prediction (Horizon 12), our method achieves the best MAE of 22.18 and RMSE of 39.43, highlighting its robustness in modeling complex long-range dependencies. On even larger traffic datasets GBA, RAST still surpasses baselines across all metrics and horizons, outperforming strong baselines like DSTAGNN and RPMixer. These results demonstrate the scalability of RAST to model complex spatio-temporal dependencies in large-scale traffic networks.

## 4.3 Ablation Study (RQ2)

| Methods | MAE | RMSE | MAPE (%) |
|---|---|---|---|
| RAST (Full) | **19.52** | **32.73** | **8.23** |
| w/o Fusion Query | 24.53$_{\downarrow 25.6\%}$ | 37.76$_{\downarrow 15.4\%}$ | 11.76$_{\downarrow 42.9\%}$ |
| w/o ST-Retriever | 21.71$_{\downarrow 11.2\%}$ | 34.58$_{\downarrow 5.7\%}$ | 10.04$_{\downarrow 22.0\%}$ |
| w/o Spatial Encoder | 22.88$_{\downarrow 17.2\%}$ | 35.93$_{\downarrow 9.8\%}$ | 10.10$_{\downarrow 22.7\%}$ |
| w/o Temporal Encoder | 23.66$_{\downarrow 21.2\%}$ | 36.90$_{\downarrow 12.7\%}$ | 10.87$_{\downarrow 32.1\%}$ |
| only Query Embed. | 23.59$_{\downarrow 20.9\%}$ | 36.81$_{\downarrow 12.5\%}$ | 10.79$_{\downarrow 31.1\%}$ |
| only Retrieval Embed. | 19.38$_{\uparrow -0.7\%}$ | 30.88$_{\uparrow -5.7\%}$ | 13.32$_{\downarrow 61.8\%}$ |
| w/o MLP Predictor | 21.67$_{\downarrow 11.0\%}$ | 34.60$_{\downarrow 5.7\%}$ | 9.52$_{\downarrow 15.7\%}$ |

Table 3: Ablation study results for different components on the PEMS07 dataset. We compare the full RAST with 7 variants. ↓ *Deg%* denotes the degradation percentage.

We conducted comprehensive ablation studies to evaluate the contribution of each component within our RAST, with results presented in Table 3. The results demonstrate the significance of each component, with performance degradations across different metrics when components are removed or modified. The query generator emerges as the most critical component, with its removal causing the most substantial performance degradation of 25.6% in MAE, 15.4% in RMSE, and 42.9% in MAPE. This dramatic decline validates our design of context-aware query generation as fundamental to the framework's effectiveness. Both spatial and temporal encoders prove indispensable, with their removal resulting in 17.2% and 21.2% MAE degradation respectively, confirming that dual-stream encoding effectively captures essential spatio-temporal dependencies.

Notably, the variant using only retrieval embeddings achieves superior MAE (19.38) and RMSE (30.88) compared to the full model while showing significant MAPE degradation (61.8%). This suggests that while retrieval alone captures overall magnitude, it may miss crucial distributional details, highlighting the importance of cross-attention-based fusion for balanced performance. The complete removal of the ST-retriever causes moderate degradation (11.2% of MAE, and 22.0% of MAPE), while the MLP predictor contributes significantly with 11.0% MAE degradation when removed. These findings collectively validate that each component contributes uniquely to capturing complex spatio-temporal dependencies.

## 4.4 Parameter Sensitivity Analysis (RQ3)

In our parameter sensitivity analysis, we focus on 4 hyperparameters: 1) the interval of retrieval, 2) the feature dimension of query embedding, 3) the dimension of retrieval embedding, and 4) the number of fusion layers. The retrieval interval demonstrates optimal performance at 10 epochs, with infrequent updates (20, 50) causing degradation due to noise introduction and pattern staleness, respectively. Query embedding dimension significantly impacts effectiveness, with performance improving dramatically from 32 to 128 dimensions, confirming the importance of sufficient representational capacity for encoding spatio-temporal patterns.

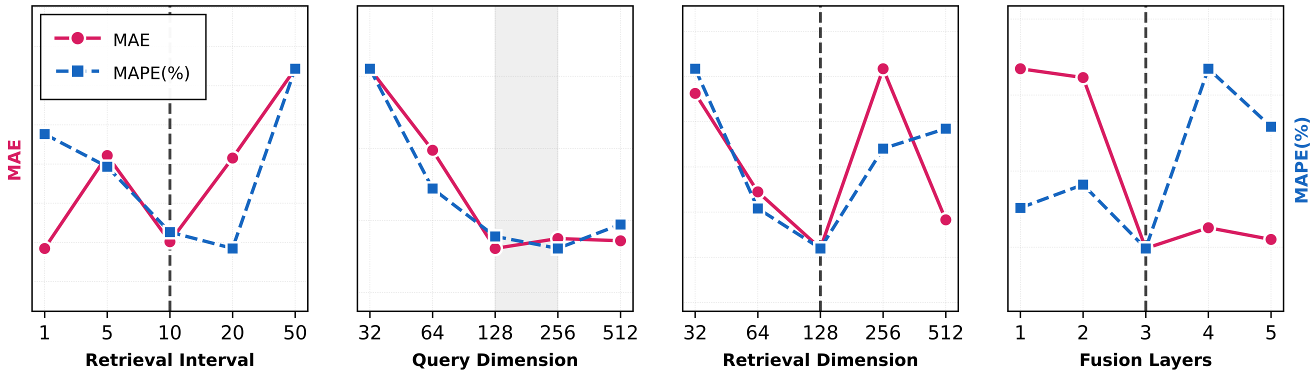

Figure 4: Hyperparameter sensitivity analysis for RAST on the PEMS04 dataset. We chose 4 crucial hyperparameters and measured results by MAE and MAPE(%). Vertical dashed lines indicate optimal values for each parameter.

| Dataset | GBA | | | SD | | |
|---|---|---|---|---|---|---|
| **Methods** | **Mem** | **Train** | **Val** | **Mem** | **Train** | **Val** |
| AGCRN | 16.39 | 619.16 | 67.63 | 5.37 | 120.14 | 12.39 |
| D2STGNN | 45.10 | 5392.56 | 830.39 | 38.36 | 1014.89 | 210.81 |
| DCRNN | 19.51 | 2654.61 | 350.11 | 9.14 | 333.83 | 57.22 |
| DGCRN | 38.71 | 2834.88 | 809.12 | 17.10 | 364.40 | 89.02 |
| GWNet | 11.24 | 1307.66 | 275.75 | 6.18 | 593.45 | 84.78 |
| STGCN | 3.40 | 852.57 | 151.46 | 2.16 | 302.95 | 66.50 |
| **RAST (Ours)** | 3.71 | **154.08** | **43.52** | 3.22 | **45.53** | **10.15** |

Table 4: Memory and efficiency comparisons on large-scale datasets. Mem: CUDA memory (GB) used, Train: training time (seconds per epoch), Val: inference time (in seconds).

The retrieval dimension demonstrates an optimal balance point at 256, while the retrieval interval is set to 10, with performance degrading at both smaller and larger values. The dimension of query embeddings significantly impacts model effectiveness, with prediction errors dramatically decreasing as resolution increases from 32 to 128, confirming the importance of the encoding processing of the raw data. The number of encoder layers shows substantial influence, with optimal performance at 3 layers (MAE: 19.19), while additional layers provide minimal improvement, suggesting that moderate network depth sufficiently extracts meaningful patterns without excessive computational costs.

These results indicate that our framework's performance gains benefit from both effective encoding of original data and appropriate dimensionality of retrieval results, while requiring only moderate update frequency for the Retrieval Store without excessive computational overhead.

### 4.5 Efficiency Analysis (RQ4)

RAST demonstrates exceptional computational efficiency, as shown in Table 4. We selected several models that demonstrated superior performance in previous experiments and evaluated them using maximum possible batch sizes (32 for SD and 16 for GBA) on a single GPU, recording three critical metrics: memory usage, training time per epoch, and inference time. This unified protocol ensures that any efficiency gains stem from algorithmic design rather than hardware advantages. While graph/attention-based models face a quadratic growing computational cost w.r.t. the number of nodes, RAST achieves the fastest training and inference speeds across both GBA and SD datasets, with training times of 154.08 and 45.53 seconds per epoch respectively, and inference times of 43.52 and 10.15 seconds. Though STGCN exhibits lower memory consumption, it requires significantly more time for both training and inference, highlighting that our model achieves better overall efficiency. Complex models like D2STGNN face severe scalability challenges on large datasets, while RAST maintains consistent performance scaling through the retrieval-augmented mechanism to capture low-predictability patterns under limited contextual capacity constraints, instead of relying on complex architectures. Through the sustainable storage and maintenance of the retrieval store, RAST maintains high performance while keeping the computational cost similar to STGCN, which verifies the effectiveness of our method under limited contextual capacity.

## 5 Conclusion

In this paper, we introduce RAST, a universal spatio-temporal forecasting framework that integrates retrieval-augmented mechanisms to tackle fine-grained spatio-temporal dependencies under limited context capacity. Comprehensive experiments across six real-world traffic datasets, including predictive benchmarks, ablation analyses, hyperparameter studies, and efficiency evaluations. The results demonstrate that RAST delivers state-of-the-art performance while retaining favorable computational efficiency. These findings highlight the promise of retrieval-augmented designs as a lightweight yet powerful complement to conventional spatio-temporal architectures, especially in large-scale and heterogeneous urban scenarios. In the future, we will continue to (i) extend RAST to broader domains such as climate modeling and electricity demand forecasting, and (ii) further optimize inference efficiency and online adaptability when interfacing with diverse pre-trained spatio-temporal graph neural networks.

# Acknowledgments

This work is mainly supported by the National Natural Science Foundation of China (No. 62402414). This work is also supported by the Guangdong Basic and Applied Basic Research Foundation (No. 2025A1515011994), Guangzhou Municipal Science and Technology Project (No. 2023A03J0011), the Guangzhou Industrial Information and Intelligent Key Laboratory Project (No. 2024A03J0628), and a grant from State Key Laboratory of Resources and Environmental Information System, and Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No. 2023B1212010007).

# References

Awad, M.; and Khanna, R. 2015. Support vector regression. In *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*, 67–80. Springer.

Bai, L.; Yao, L.; Li, C.; Wang, X.; and Wang, C. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33: 17804–17815.

Bi, K.; Xie, L.; Zhang, H.; Chen, X.; Gu, X.; and Tian, Q. 2023. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619(7970): 533–538.

Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; Van Den Driessche, G.; Lespiau, J.-B.; Damoc, B.; Clark, A.; et al. 2022. Improving language models by retrieving from trillions of tokens. *International conference on machine learning*, 2206–2240.

Box, G. E.; Jenkins, G. M.; Reinsel, G. C.; and Ljung, G. M. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

Cao, L.; Wang, B.; Jiang, G.; Yu, Y.; and Dong, J. 2025. Spatiotemporal-aware Trend-Seasonality Decomposition Network for Traffic Flow Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 11463–11471.

Challu, C.; Olivares, K. G.; Oreshkin, B. N.; Garza, F.; Mergenthaler-Canseco, M.; and Dubrawski, A. 2022. N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting. arXiv:2201.12886.

Chandra, S. R.; and Al-Deek, H. 2009. Predictions of freeway traffic speeds and volumes using vector autoregressive models. *Journal of Intelligent Transportation Systems*, 13(2): 53–72.

Chavhan, S.; and Venkataram, P. 2020. Prediction based traffic management in a metropolitan area. *Journal of traffic and transportation engineering (English edition)*, 7(4): 447–466.

Chen, Y.; Lv, Y.; Wang, F.; and Zhang, X. 2021. Multimodal fusion learning for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Chen, Y.; Segovia-Dominguez, I.; Coskunuzer, B.; and Gel, Y. 2022. TAMP-S2GCNets: coupling time-aware multipersistence knowledge representation with spatio-supra graph convolutional networks for time-series forecasting. In *International conference on learning representations*.

Devlin, J. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Douze, M.; Guzhva, A.; Deng, C.; Johnson, J.; Szilvasy, G.; Mazaré, P.-E.; Lomeli, M.; Hosseini, L.; and Jégou, H. 2024. The Faiss library.

Fang, Z.; Long, Q.; Song, G.; and Xie, K. 2021. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 364–373.

Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; and Wang, H. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Guo, S.; Lin, Y.; Feng, N.; Song, C.; and Wan, H. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 922–929.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Huang, R.; Huang, C.; Liu, Y.; Dai, G.; and Kong, W. 2020. LSGCN: Long short-term traffic prediction with graph convolutional networks. In *IJCAI*, volume 7, 2355–2361.

Izacard, G.; and Grave, E. 2021. Leveraging passage retrieval with generative models for open domain question answering. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, 874–880.

Jiang, R.; Yin, D.; Wang, Z.; Wang, Y.; Deng, J.; Liu, H.; Cai, Z.; Deng, J.; Song, X.; and Shibasaki, R. 2021. Dl-traff: Survey and benchmark of deep learning models for urban traffic prediction. In *Proceedings of the 30th ACM international conference on information & knowledge management*, 4515–4525.

Jiang, Z. 2023. Deep Learning for Spatiotemporal Big Data: A Vision on Opportunities and Challenges. *arXiv preprint arXiv:2310.19957*.

Jin, G.; Liang, Y.; Fang, Y.; Shao, Z.; Huang, J.; Zhang, J.; and Zheng, Y. 2023. Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Jin, K.; Wi, J.; Lee, E.; Kang, S.; Kim, S.; and Kim, Y. 2021. TrafficBERT: Pre-trained model with large-scale data for long-range traffic flow forecasting. *Expert Systems with Applications*, 186: 115738.

Jin, M.; Koh, H. Y.; Wen, Q.; Zambon, D.; Alippi, C.; Webb, G. I.; King, I.; and Pan, S. 2024a. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jin, M.; Zhang, Y.; Chen, W.; Zhang, K.; Liang, Y.; Yang, B.; Wang, J.; Pan, S.; and Wen, Q. 2024b. Position: What Can Large Language Models Tell Us about Time Series Analysis. In *Forty-first International Conference on Machine Learning*.

Johnson, J.; Douze, M.; and Jégou, H. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3): 535–547.

Kandpal, N.; Deng, H.; Roberts, A.; Wallace, E.; and Raffel, C. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, 15696–15707. PMLR.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Lan, S.; Ma, Y.; Huang, W.; Wang, W.; Yang, H.; and Li, P. 2022. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International conference on machine learning*, 11906–11917. PMLR.

Lea, C.; Flynn, M. D.; Vidal, R.; Reiter, A.; and Hager, G. D. 2017. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 156–165.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474.

Li, M.; and Zhu, Z. 2021. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 4189–4196.

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

Liang, Y.; Ke, S.; Zhang, J.; Yi, X.; and Zheng, Y. 2018. Geoman: Multi-level attention networks for geo-sensory time series prediction. In *IJCAI*, volume 2018, 3428–3434.

Liu, H.; Dong, Z.; Jiang, R.; Deng, J.; Deng, J.; Chen, Q.; and Song, X. 2023a. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 4125–4129.

Liu, X.; Xia, Y.; Liang, Y.; Hu, J.; Wang, Y.; Bai, L.; Huang, C.; Liu, Z.; Hooi, B.; and Zimmermann, R. 2023b. LargeST: A Benchmark Dataset for Large-Scale Traffic Forecasting. In *Advances in Neural Information Processing Systems*.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023c. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

Lütkepohl, H. 2005. *New introduction to multiple time series analysis*. Springer Science & Business Media.

Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; and Wang, F.-Y. 2014. Traffic flow prediction with big data: A deep learning approach. *Ieee transactions on intelligent transportation systems*, 16(2): 865–873.

Petroni, F.; Rocktäschel, T.; Lewis, P.; Bakhtin, A.; Wu, Y.; Miller, A. H.; and Riedel, S. 2019. Language models as knowledge bases? *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2463–2473.

Ram, O.; Levine, Y.; Dalmedigos, I.; Muhlgay, D.; Shashua, A.; Leyton-Brown, K.; and Shoham, Y. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11: 1316–1331.

Roberts, A.; Raffel, C.; and Shazeer, N. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

Shao, Z.; Zhang, Z.; Wang, F.; Wei, W.; and Xu, Y. 2022a. 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*, 4454–4458.

Shao, Z.; Zhang, Z.; Wang, F.; and Xu, Y. 2022b. 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*, 1567–1577.

Shao, Z.; Zhang, Z.; Wei, W.; Wang, F.; Xu, Y.; Cao, X.; and Jensen, C. S. 2022c. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *Proceedings of the VLDB Endowment*, 15(11): 2733–2746.

Shi, W.; Min, S.; Yasunaga, M.; Seo, M.; James, R.; Lewis, M.; Zettlemoyer, L.; and Yih, W.-t. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28.

Song, C.; Lin, Y.; Guo, S.; and Wan, H. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 914–921.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, S.; Cao, J.; and Philip, S. Y. 2020. Deep learning for spatio-temporal data mining: A survey. *IEEE transactions on knowledge and data engineering*, 34(8): 3681–3700.

Wang, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J. Y.; and Zhou, J. 2024a. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*.

Wang, X.; Ma, Y.; Wang, Y.; Jin, W.; Wang, X.; Tang, J.; Jia, C.; and Yu, J. 2020. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020*, 1082–1092.

Wang, X.; Wang, Z.; Wang, E.; and Sun, Z. 2024b. Spatial-temporal knowledge distillation for lightweight network traffic anomaly detection. *Computers & Security*, 137: 103636.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. 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*, 753–763.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*.

Yan, Y.; Wen, H.; Zhong, S.; Chen, W.; Chen, H.; Wen, Q.; Zimmermann, R.; and Liang, Y. 2024. Urbanclip: Learning text-enhanced urban region profiling with contrastive language-image pretraining from the web. In *Proceedings of the ACM on Web Conference 2024*, 4006–4017.

Yeh, C.-C. M.; Fan, Y.; Dai, X.; Saini, U. S.; Lai, V.; Aboagye, P. O.; Wang, J.; Chen, H.; Zheng, Y.; Zhuang, Z.; et al. 2024. Rpmixer: Shaking up time series forecasting with random projections for large spatial-temporal data. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3919–3930.

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

Yuan, Y.; Ding, J.; Feng, J.; Jin, D.; and Li, Y. 2024. Unist: a prompt-empowered universal model for urban spatio-temporal prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4095–4106.

Zhang, J.; Zheng, Y.; and Qi, D. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-first AAAI conference on artificial intelligence*.

Zheng, C.; Fan, X.; Wang, C.; and Qi, J. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 1234–1241.

Zheng, Y.; Capra, L.; Wolfson, O.; and Yang, H. 2014. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3): 1–55.

Zhou, T.; Niu, P.; Sun, L.; Jin, R.; et al. 2024. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36.

Zhou, Z.; Hu, J.; Wen, Q.; Kwok, J. T.; and Liang, Y. 2025. Multi-Order Wavelet Derivative Transform for Deep Time Series Forecasting. arXiv preprint arXiv:2505.11781.

Zhu, K.; Luo, Y.; Xu, D.; Wang, R.; Yu, S.; Wang, S.; Yan, Y.; Liu, Z.; Han, X.; Liu, Z.; et al. 2024. Rageval: Scenario specific rag evaluation dataset generation framework. *arXiv preprint arXiv:2408.01262*.