Unified Spatio-Temporal Tokens are Bases for Generalizable Traffic Forecasting

Yujun Chen*
yujun.chen@zju.edu.cn
Zhejiang University
Hangzhou, China

Shihao Tu*
tushihao@supcon.com
SUPCON Technology Co., Ltd.
Hangzhou, China

Wenyue Ding
dingwenyue@supcon.com
SUPCON Technology Co., Ltd.
Hangzhou, China

Yicheng Lu
yicheng.lu@zju.edu.cn
Zhejiang University
Hangzhou, China

Qingkai Ren
qingkairen@zju.edu.cn
Zhejiang University
Hangzhou, China

Yangjie Zheng
yangjie.zheng@zju.edu.cn
Zhejiang University
Hangzhou, China

Yang Yang†
yangya@zju.edu.cn
Zhejiang University
Hangzhou, China

Abstract

Traffic forecasting plays a crucial role in real-world applications such as traffic management and urban planning. Recent studies have mainly focused on spatio-temporal graph neural networks (STGNNs) and attention-based methods, which have shown promising results. Nevertheless, both approaches model spatial information implicitly, which limits their ability to generalize across different traffic networks. In this paper, we propose Spatio-Temporal Unified Network (STUNet), a framework to explicitly encode spatial features into unified representations and integrate them with temporal information effectively. To obtain spatial representations explicitly, we design a spatial tokenizer that segments the adjacency matrix of the relation graph into patches to serve as spatial tokens. Furthermore, to effectively integrate spatial and temporal representations, we introduce query-aggregate attention, which simulates the process of tracing upstream and downstream nodes and aggregating their information, thereby capturing complex spatio-temporal dependencies. Extensive experiments on traffic benchmarks demonstrate that STUNet achieves generalization across different traffic networks with competitive performance. Code is available at https://github.com/JimmyChen6/STUNet.

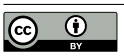
CCS Concepts

• **Computing methodologies** → **Temporal reasoning**; • **Information systems** → **Geographic information systems**.

Keywords

traffic forecasting, Transformer, explicit graph modeling

*Both authors contributed equally to this research.
†Corresponding Authors



KDD 2026, Jeju Island, Republic of Korea.

ACM ISBN 979-8-4007-2259-2/2026/08
https://doi.org/10.1145/3770855.3817746

ACM Reference Format:
Yujun Chen, Shihao Tu, Wenyue Ding, Yicheng Lu, Qingkai Ren, Yangjie Zheng, and Yang Yang. 2026. Unified Spatio-Temporal Tokens are Bases for Generalizable Traffic Forecasting. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD 2026), August 9–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3770855.3817746

1 Introduction

Traffic forecasting is a critical task with significant practical implications, offering invaluable support for intelligent transportation systems, such as optimizing resource allocation in command and dispatch centers and facilitating the development of smart cities [47]. Traffic data is fundamentally a multivariate time series collected by road-side sensors, characterized by complex spatial relationships. A key characteristic of this data is its strong spatial dependency [18], primarily because traffic flow propagates sequentially along the underlying road topology (i.e., from upstream to downstream). Furthermore, since road networks are typically constructed from a combination of fundamental components (e.g., overpasses and roundabouts), the observed spatial dependency patterns are often transferable across different geographical regions. For example, as shown in Figure 1, Y-intersections are typical fundamental components of traffic networks and the behavior of diverging and merging can be generalized from one to the other. This generalizability of spatial information holds significant value for building robust and scalable forecasting models.

Extensive research has been conducted to achieve accurate traffic flow forecasting. Early time-series forecasting models have evolved ranging from statistic methods [3, 21] to data-driven methods [27, 48]. With the rise of transformer [37], a plethora of attention-based time-series forecasting models have emerged [28, 39, 40, 52, 53]. Although these models have achieved promising results in general forecasting tasks, univariate models are inherently limited in their ability to capture complex spatial information, which leads to suboptimal performance when applied to traffic data.

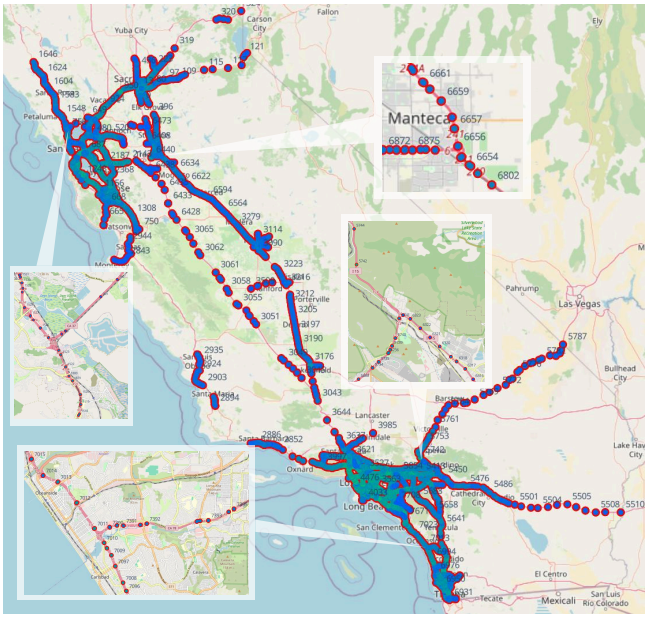


Figure 1: Road networks are typically constructed from a combination of fundamental components. Here are some Y-intersections from different parts of a huge network.

To capture the spatial information of the data, researchers have turned to multivariate time-series models. Inspired by graph neural networks (GNNs) [7, 12, 15, 20, 34], spatio-temporal graph neural networks(STGNNs) utilize a message-passing mechanism to aggregate information from neighboring sensors [19]. Here spatial information serves as the medium through which temporal information is propagated, so they're implicitly represented. However, the relationships between sensors may vary over time, whereas GNNs themselves can only accept a given graph and fail to capture such dynamic changes. Attention-based methods construct spatial relationships using the inner product of sensor representations, thereby capturing dynamic sensor relationships; in these methods, spatial information is implicitly modeled in the form of attention scores. Furthermore, both STGNNs and attention-based methods introduce a spatial complexity of $O(N^2)$ for spatial information. Some approaches choose to maintain node-wise spatial embeddings to implicitly model spatial information [32], which reduces the spatial complexity of spatial information to $O(N)$.

However, these methods all model spatial information implicitly, where spatial and temporal information are fully coupled during training. Such implicit modeling will lead to the learning of spatial information being inevitably influenced by temporal dynamics. For traffic data, spatial information should ideally be determined by the underlying road structure rather than the observed values from sensors. The coupled training paradigm in existing methods allows temporal information to interfere with spatial learning, and this erroneous causal relationship compromises the generalizability of the learned spatial information. In contrast, explicitly modeling spatial information can decouple spatial learning from the overall training process, ensuring that spatial representations remain unaffected by temporal fluctuations and thus achieving stronger generalizability. Nevertheless, the number and distribution of sensors in traffic data is not fixed, making it a significant challenge to encode spatial information into a unified representation without losing structural details. Furthermore, the representations derived from explicit spatial modeling and those from temporal information do not share the same semantic space, posing a major difficulty in achieving effective interaction between the two. In summary, explicit modeling of spatial information faces the following challenges:

- How to explicitly map spatial information into a unified representation without losing spatial structural information.
- How to effectively fuse spatial and temporal representations that are semantically misaligned.

To address the challenges above, we propose the Spatio-Temporal Unified Network (STUNet). The framework of STUNet is illustrated in the model diagram 2. The core objective of this model is to explicitly model spatial information and effectively fuse it with temporal information. Inspired by ViT [6], our spatial tokenizer partitions the adjacency matrix into spatial patches and maps them into spatial tokens. These spatial tokens preserve the information of the adjacency matrix and avoid the high complexity associated with point-wise modeling. Following PatchTST [28], the temporal tokenizer segments the raw sequences into temporal patches and maps them into temporal tokens to retain local information. Based on these spatial and temporal tokens, we propose query-aggregate attention to effectively fuse their information. It first performs cross-attention between spatial and temporal tokens to derive the relative positional relationships between sensors, and then applies self-attention to the temporal tokens to aggregate causal sensor information. Notably, since spatial and temporal tokens differ in their spatial arrangement and their positional relationships are interconnected, we apply two sets of positional encodings within the query-aggregate attention to preserve their respective positional information. Furthermore, to ensure that spatial modeling remains unaffected by temporal information, we employ a pre-training strategy for both the spatial and temporal tokenizers. After pre-training, the parameters of the spatial tokenizer are frozen to ensure that spatial tokens accurately reflect the spatial structure, thereby achieving generalizability in spatial information modeling.

In summary, the main contributions of our work are as follows:

- We propose an explicit tokenization strategy of spatial information in traffic forecasting, addressing the challenge of generalizability in spatial information modeling. Our spatial tokenizer partitions the adjacency matrix into patches, thereby transforming spatial information into a unified representation.
- We propose query-aggregate attention, which utilizes carefully designed positional encodings to ensure the validity of relative positions during information fusion, effectively achieving the integration of spatial and temporal information.
- Building upon the aforementioned contributions, we propose STUNet, which explicitly maps spatial and temporal information into unified representations and performs effective information fusion. Extensive experiments on traffic forecasting benchmarks demonstrate that STUNet achieves generalization across different networks with competitive performance.

2 Related Work

2.1 Implicit Graph Modeling

Graphs are widely accepted by researchers as a powerful candidate for representing spatial relations. To date, most spatio-temporal forecasting methods have utilized graphs as a medium for information propagation, thereby implicitly integrating spatial information to assist in forecasting. These methods can be broadly categorized into two types: spatio-temporal graph neural networks (STGNNs) and Transformers [19, 37]. The former leverages GNNs to utilize given or learnable graphs for information passing, while the latter employs attention mechanisms to achieve graph construction and information propagation.

STGNNs Inspired by the success of graph neural networks (GNNs) [12, 20, 38, 43], researchers have extensively explored STGNNs. Early STGNNs utilized a given prior graph [DCRNN, STGCN] to perform information propagation using static spatial relationships [24, 46]. To capture spatial dependencies in data without a prior graph, researchers have proposed various adaptive methods to obtain learnable graphs. These methods primarily maintain node-wise embeddings and construct graphs using their inner products [1, 5, 41, 42]. Recognizing that spatial dependencies may vary over time, researchers have subsequently proposed several STGNNs based on dynamic graphs [2, 11, 17]. To simultaneously capture both local and global information, some paradigms have been introduced that utilize multiple graphs concurrently [10, 23]. PowerPM provides a hierarchical view for GNNs [36]. Furthermore, to mitigate the high computational complexity of STGNNs, certain methods have adopted LoRA or linear approximations to enhance efficiency [13, 30].

Transformers Attention has achieved significant success in fields such as NLP and CV [6, 37]. Since the computation process of attention can be viewed as a simulation of spatial relationships, spatio-temporal forecasting models based on Transformers have attained considerable success [29, 49, 51]. PDFormer utilizes both geographic and semantic masks to simultaneously capture short-range and long-range relationships. To efficiently process spatial information between sensors [16], STWave samples active sensors into coefficient sequences [9]. To reduce computational latency, ASTNet proposes an asynchronous mechanism for spatial and temporal learning [35].

Furthermore, to combine the advantages of the aforementioned methods, TESTAM utilizes MoE to route between channel-independent learnable graphs and attention mechanisms [22]. Although STGNNs and Transformers have achieved promising results, their paradigm of implicit graph modeling makes them more dependent on temporal features. This leads to a lack of robust perception of structural information in the spatial dimension, which in turn limits their generalizability across different traffic networks.

2.2 Graph-free Spatial Modeling

Graph-based algorithms often suffer from high computational complexity; consequently, a subset of research avoids using graphs to model spatial relationships. STID directly preserves spatial information through node-wise embeddings and uses only an MLP as the backbone, serving as a highly efficient and competitive baseline [32]. CMuST introduces cross-attention between spatial embeddings and context to achieve the fusion of contextual information [44]. PatchSTG performs sensor sampling via KDTree and employs depth and breadth attention to efficiently learn spatio-temporal information [8]. However, these graph-free methods are predominantly based on node-wise embeddings. Whenever the scenario changes, these embeddings must be re-trained, meaning they lack the capability for generalization across different traffic networks.

2.3 Explicit Graph Modeling in Other Fields

Explicit graph tokenization has been explored in graph and computer vision communities. Graphormer introduces centrality encoding, spatial encoding, and edge encoding to inject graph structural information into the Transformer architecture, achieving state-of-the-art performance on graph representation benchmarks [45]. GraphGPS proposes a modular framework that combines local message-passing with global attention, demonstrating that explicit positional/structural encodings are essential for building general and powerful graph Transformers [31]. SAT extracts k-hop subgraph representations as structural tokens before computing attention, enabling structure-aware self-attention that captures node-level structural similarity [4]. In the vision domain, ViT explicitly partitions images into fixed-size patches as visual tokens [6], while MAE further leverages this patch-based representation for self-supervised pre-training via masked autoencoding [14]. These works demonstrate that explicitly encoding structural information into unified token representations is a promising paradigm.

To address the lack of cross-network generalization in previous studies, we propose STUNet, a model that directly performs explicit modeling on relationship graphs. STUNet partitions the graph into equal-length patches and transforms them into aligned spatial tokens. These tokens are derived directly from the graph and possess semantics independent of temporal information, thereby enabling generalization across different traffic networks.

3 Preliminaries

Multivariate Time Series Data. Multivariate time series data is made up of multiple univariate time series. We denote $\mathbf{X}^i \in \mathbb{R}^T$ as the univariate time series of sensor i, where T is the number of time steps. The complete multivariate time series data with N sensors can be represented as $\mathbf{X} \in \mathbb{R}^{T \times N}$.

Relation Graph. The relation graph of the multivariate time series data is represented as $\mathbf{A} \in \mathbb{R}^{N \times N}$, where $\mathbf{A}_{ij} = 1$ indicates a connection between node i and node j, and $\mathbf{A}_{ij} = 0$ otherwise.

Spatial-Temporal Forecasting. Given historical multivariate time series data $\mathbf{X} \in \mathbb{R}^{H \times N}$, the goal of spatial-temporal forecasting is to predict the future F time steps of the multivariate time series, denoted as $\mathbf{Y} \in \mathbb{R}^{F \times N}$.

4 Methodology

In this section, we introduce the overall framework of our proposed model, Spatio-Temporal Unified Network(STUNet). Our model is designed to maintain high efficiency and generalizability in large-scale traffic forecasting tasks. STUNet consists of four main components: (1) a spatial encoder that transforms spatial information (relationship graphs) into spatial tokens; (2) a temporal tokenizer that converts time-series data into temporal tokens aligned with

the spatial tokens; (3) a query-aggregate attention mechanism used to facilitate interaction between temporal and spatial tokens to capture complex spatio-temporal dependencies; and (4) a prediction head for generating the final forecasting values. Besides, to decompose spatial information and temporal information modeling, we utilized a two-stage training strategy.

4.1 Framework

4.1.1 Spatial Tokenizer. Explicitly modeling the structure of a relationship graph necessitates a robust graph encoding mechanism that preserves spatial structural properties without information loss. Since the number of sensors in the data is not fixed, direct mapping of the relationship graph is infeasible. Furthermore, assigning a token to every position on the relationship graph would result in catastrophic computational complexity. Fortunately, inspired by ViT [6], we propose a spatial tokenizer that partitions the relationship graph into several patches. Since these patches have aligned dimensions and uniform sizes, they can be processed directly. Compared to point-wise processing, patch-based processing significantly reduces complexity while maintaining information integrity. Given the adjacency matrix of the relationship graph $\mathbf{A} \in \mathbb{R}^{N\times N}$, we partition it into non-overlapping patches: $\mathbf{A}_p \in \mathbb{R}^{(P_s)^2\times(L_s)^2}$, where $(P_s)^2 = \lceil \frac{N}{L_s} \rceil^2$ is the total number of patches, L_s is the stride corresponding to each patch. If the number of sensors N is not divisible by L_s, the borders of the adjacency matrix are padded with zeros. Subsequently, we employ an MLP: $\mathbb{R}^{(L_s)^2} \mapsto \mathbb{R}^d$ to transform these patches into spatial tokens $\mathbf{E}_s \in \mathbb{R}^{(P_s)^2\times d}$. In summary, our spatial tokenizer effectively preserves spatial information while achieving low computational complexity by partitioning the adjacency matrix of the relationship graph into patches. In addition, the spatial tokenizer is pre-trained as mentioned in Training Strategy. For the details of the implementation of STUNet, please refer to Appendix B.

4.1.2 Temporal Tokenizer. Following previous works [28], we split input multivariate time series data $\mathbf{X} \in \mathbb{R}^{H\times N}$ into patches $\mathbf{X}_p \in \mathbb{R}^{(N\times P_t)\times L_t}$, where $P_t = \lfloor \frac{H-L_t+1}{S_t} \rfloor + 1$ is the number of patches of one sensor, L_t is the length of each patch and S_t is the stride. Zero padding is adopted if necessary. Then we use a MLP to map $\mathbf{X}_p$ into $\mathbf{E}_x \in \mathbb{R}^{(N\times P_t)\times d_x}$. Besides, due to the periodic patterns in traffic data, we also add temporal embedding including the number of days in a week and the number of timeslices in a day, i.e. $\mathbf{E}_{tod} \in \mathbb{R}^{(N\times P_t)\times d_{tod}}$ and $\mathbf{E}_{dow} \in \mathbb{R}^{(N\times P_t)\times d_{dow}}$. Finally, we concatenate them to get the temporal tokens $\mathbf{E}_t = \mathbf{E}_x \| \mathbf{E}_{tod} \| \mathbf{E}_{dow} \in \mathbb{R}^{(N\times P_t)\times d}$, where $d = d_x + d_{tod} + d_{dow}$ is the dimension of temporal tokens. The temporal tokenizer is pre-trained as mentioned in Training Strategy as well.

4.1.3 Query-Aggregate Attention. Through the spatial tokenizer and temporal tokenizer, we obtain aligned spatial tokens $\mathbf{E}_s$ and temporal tokens $\mathbf{E}_t$, respectively. In this section, we describe the interactive fusion of temporal and spatial information. Since these two sets of tokens originate from different modalities, they possess distinct informational meanings and exhibit different properties regarding relative position calculations. Consequently, they cannot be simply mixed for attention computation. To address this issue, we propose Query-Aggregate Attention, which decouples the complex operation of fusing spatio-temporal information into two simple and meaningful operations: query and aggregation. The backbone of our STUNet is constructed by stacking multiple Query-Aggregate Attention modules. For convenience, we denote the inputs to the i-th Query-Aggregate Attention module as $\mathbf{E}_s^{i-1}$ and $\mathbf{E}_t^{i-1}$. We have $\mathbf{E}_s^{i-1} = \mathbf{E}_s$ for every layer and $\mathbf{E}_t^0 = \mathbf{E}_t$ for the first layer.

Query Attention. In query attention, we simulate the query operation by identifying the positions of nodes that are beneficial for the current node's prediction. For the inputs $\mathbf{E}_s^{i-1}$ and $\mathbf{E}_t^{i-1}$, we obtain the query, key, and value vectors through the following mappings:

$$\mathbf{Q}_q^i = \text{TemporalPE}_\text{q}\left(\mathbf{E}_t^{i-1}\mathbf{W}_{\mathbf{Q}_q}^i\right) \tag{1}$$

$$\mathbf{K}_q^i = \text{SpatialPE}_\text{q}\left(\mathbf{E}_s^{i-1}\mathbf{W}_{\mathbf{K}_q}^i\right) \tag{2}$$

$$\mathbf{V}_q^i = \mathbf{E}_s^{i-1}\mathbf{W}_{\mathbf{V}_q}^i \tag{3}$$

where $\mathbf{W}_{\mathbf{Q}_q}^i, \mathbf{W}_{\mathbf{K}_q}^i, \mathbf{W}_{\mathbf{V}_q}^i \in \mathbb{R}^{d\times d}$ are learnable parameters. Considering that the positional information of the query and key vectors is different, we design the following RoPE strategy [33]: Let $\mathbf{Z}_t^{n,p}$ be the embedding corresponding to the p-th patch of sensor n. We divide it equally into two parts, $\mathbf{Z}_{t,f}^{n,p}$ and $\mathbf{Z}_{t,b}^{n,p}$:

$$\text{TemporalPE}_\text{q}\left(\mathbf{Z}_t^{n,p}\right) = \text{RoPE}\left(\mathbf{Z}_{t,f}^{n,p}, n\right) \| \text{RoPE}\left(\mathbf{Z}_{t,b}^{n,p}, n\right) \tag{4}$$

Let $\mathbf{Z}_s^{j,k}$ be the spatial embedding at row j and column k. We divide it equally into two parts, $\mathbf{Z}_{s,f}^{j,k}$ and $\mathbf{Z}_{s,b}^{j,k}$:

$$\text{SpatialPE}_\text{q}\left(\mathbf{Z}_s^{j,k}\right) = \text{RoPE}\left(\mathbf{Z}_{s,f}^{j,k}, j\right) \| \text{RoPE}\left(\mathbf{Z}_{s,b}^{j,k}, k\right) \tag{5}$$

Since query attention involves the interaction between temporal tokens and spatial tokens, only the sensor index information corresponding to both is relevant here. During the inner product operation, the first half of the tokens correspond to querying the relative row positions of temporal tokens in the relationship graph, while the second half corresponds to querying the relative column positions. For spatial tokens, the two parts of PE mark the row and column of them, which preserve their 2D positional information. For temporal tokens, two parts of PE enable them to find their neighbors(both source and target) through relative position with spatial tokens. Thus, our PE mechanism preserves the computational properties of RoPE, thereby achieving two-dimensional relative position awareness of temporal tokens within the spatial tokens. Subsequently, we utilize the attention mechanism to complete the query operation:

$$\mathbf{H}^i = \text{Softmax}\left(\frac{\mathbf{Q}_q^i \mathbf{K}_q^{i\,T}}{\sqrt{d}}\right)\mathbf{V}_q^i \tag{6}$$

Here, each token in $\mathbf{H}^i$ represents the corresponding temporal token having aggregated the positional information of its upstream and downstream sensors, thus completing the query operation.

Aggregate Attention. In query attention, we obtain $\mathbf{H}^i$, which contains the temporal information of the current sensor and the positional information of its upstream and downstream sensors. In this stage, we perform the aggregation operation based on this

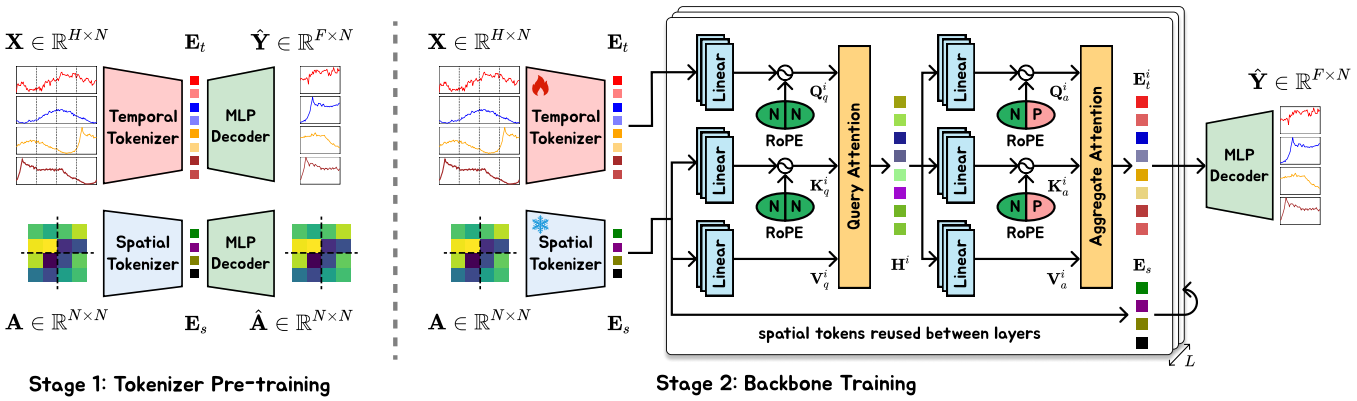


Figure 2: An illustration of STUNet framework

information. We obtain the query, key, and value vectors through the following mappings:

$$\mathbf{Q}_a^i = \text{TemporalPE}_\text{a}\left(\mathbf{H}^i\mathbf{W}_{\mathbf{Q}_a}^i\right) \tag{7}$$

$$\mathbf{K}_a^i = \text{TemporalPE}_\text{a}\left(\mathbf{H}^i\mathbf{W}_{\mathbf{K}_a}^i\right) \tag{8}$$

$$\mathbf{V}_a^i = \mathbf{H}^i\mathbf{W}_{\mathbf{V}_a}^i \tag{9}$$

where $\mathbf{W}_{\mathbf{Q}_a}^i, \mathbf{W}_{\mathbf{K}_a}^i, \mathbf{W}_{\mathbf{V}_a}^i \in \mathbb{R}^{d\times d}$ are learnable parameters. In aggregate attention, each token needs to attend to the temporal information of the same sensor while also aggregating temporal information from relevant sensors based on the learned positional relationships. Therefore, we design the following positional encoding: Let $\mathbf{Z}_t^{n,p}$ be the embedding corresponding to the p-th patch of sensor n. We divide it equally into two parts, $\mathbf{Z}_{t,f}^{n,p}$ and $\mathbf{Z}_{t,b}^{n,p}$:

$$\text{TemporalPE}_\text{a}\left(\mathbf{Z}_t^{n,p}\right) = \text{RoPE}\left(\mathbf{Z}_{t,f}^{n,p}, n\right) \,\|\, \text{RoPE}\left(\mathbf{Z}_{t,b}^{n,p}, p\right) \tag{10}$$

During the inner product operation, the first half of the token perceives the positional information of the sensors and aggregates temporal information from relevant sensors, while the second half perceives the time lag relationships between tokens across time steps. The aggregation operation is as follows:

$$\mathbf{E}_t^i = \text{Softmax}\left(\frac{\mathbf{Q}_a^i \mathbf{K}_a^{i\,T}}{\sqrt{d}}\right)\mathbf{V}_a^i \tag{11}$$

This completes one iteration of Query-Aggregate Attention. Through meticulous task decomposition and clever positional encoding, we achieve efficient fusion of temporal and spatial information.

4.1.4 Projection Head. After L layers of Query-Aggregate Attention, we obtain the original spatial tokens $\mathbf{E}_s \in \mathbb{R}^{(P_s)^2\times d}$ and the output tokens $\mathbf{E}_t^L \in \mathbb{R}^{(N\times P_t)\times d}$ that integrate spatio-temporal information. Since the spatial tokens do not contain temporal information, we utilize only the output tokens for downstream forecasting tasks. Specifically, each output token corresponds to the information aggregated from the P_t-th patch of the N-th sensor. We concatenate the tokens for each sensor into $\mathbf{E}_o \in \mathbb{R}^{N\times(P_t\times d)}$ and pass them through an MLP to obtain the predicted values:

$$\hat{\mathbf{Y}} = \mathbf{E}_o\mathbf{W}_\mathbf{Y} + \mathbf{b}_\mathbf{Y} \in \mathbb{R}^{N\times F} \tag{12}$$

where $\mathbf{W}_\mathbf{Y} \in \mathbb{R}^{(P_t\times d)\times F}$ and $\mathbf{b}_\mathbf{Y} \in \mathbb{R}^{N\times F}$ are learnable parameters. Finally, we optimize STUNet using the prediction loss:

$$\mathcal{L} = ||\hat{\mathbf{Y}} - \mathbf{Y}|| \tag{13}$$

4.2 Training Strategy

The quality of the tokens output by the spatial and temporal tokenizers is critical to the overall performance of the model. If STUNet were trained in an end-to-end paradigm, spatial information modeling would be influenced by temporal information, thus reducing the generalizability of STUNet. Besides, such a paradigm would lead to mutual interference between spatial and temporal tokens and require additional effort to balance their respective scales, often failing to yield optimal training results. Thus, we adopt a two-stage training paradigm: first, the spatial and temporal tokenizers are trained independently using autoencoder; subsequently, the pre-trained tokenizers are integrated into the backbone for joint training. Note that the same set of sensors may correspond to different adjacency matrix since the order of them can be different, to handle this, during the first stage of training the spatial tokenizer, we randomly shuffle the node indices of the input relationship graph to generate diverse adjacency matrices. This would also make spatial tokenizer more robust. Moreover, since spatial information has been modeled effectively in the first stage, we utilize the prediction loss only to optimize the model and freeze the parameters of the spatial tokenizer during the second stage. Thus, the representation of spatial information would not be influenced by temporal information, which can achieve greater generalizability.

5 Experiments

In this section, to evaluate our proposed STUNet comprehensively, we conduct extensive experiments to address the following research questions:

- **RQ1.** How does the generalization ability of our STUNet perform when comparing with other approaches?
- **RQ2.** Compared to existing approaches in large-scale traffic forecasting, how does STUNet perform?
- **RQ3.** Are the main components within STUNet effective in enhancing the model's performance?

- **RQ4.** What are the requirements of STUNet regarding its hyperparameters?
- **RQ5.** How does the efficiency of STUNet compare with other existing methods?

5.1 Setups

Datasets Following LargeST [25], we conduct experiments on on four traffic flow datasets SD, GBA, GLA, CA and three traffic velocity datasets METR-LA, PEMS-BAY, SZ-TAXI [24, 25, 50]. We split each dataset into train, validation, test sets with a ratio of 6:2:2 for SD, GBA, GLA, CA, SZ-TAXI and 7:1:2 for others. We utilize 24 continuous time slices as a sample with the historical window of 12 and the future window of 12. For detailed statistics of these datasets, please refer to appendix A.

Evaluation metrics We utilize diverse evaluation criteria from performance and efficiency aspects for a comprehensive comparison. For the performance aspect: we utilize three commonly adopted numerical metrics to assess the performance of predicted traffic time series, i.e., mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). For the efficiency aspect: the measurement of the model's efficiency is based on the wall-clock time, and the memory needs of models are revealed by the batch size in the training phase.

5.2 Zero-shot performance(RQ1)

To evaluate the generalizability of our proposed STUNet, we selected three non-overlapping flow datasets SD, GBA, GLA and three velocity datasets METR-LA, PEMS-BAY, SZ-TAXI. For all methods, we conducted testing on two datasets after the model had converged on the third, thereby obtaining their zero-shot performance, which serves as a direct reflection of their generalizability. In this experiment, we selected six baseline methods: (i) for univariate models, we chose DLinear [48], PatchTST [28], and STID [32]; (ii) for multivariate models, we selected iTransformer [26], STGCN [46], STWave [9], and PatchSTG [8]. Specifically, for graph-free approaches which use spatial embeddings, we remove these node-wise embeddings to perform zero-shot experiments.Table 1 and 2 presents the zero-shot capabilities of all baseline methods alongside our STUNet. We observe the following results:

Multivariate models exhibit stronger generalizability compared to univariate models. As shown in Table 1 and 2, multivariate models generally achieve lower error metrics across all indicators than univariate models. This is attributed to the fact that multivariate models integrate spatial information, which is a critical factor influencing the fusion of temporal information across different sensors in traffic data. The inclusion of such spatial information significantly enhances the model's generalizability.

Explicit spatial modeling demonstrates superior generalizability over implicit graph structure modeling. STUNet achieves the best performance across all scenarios and metrics in Table 1 and 2, surpassing all methods that rely on implicit spatial modeling. This superiority stems from the fact that implicit modeling paradigms often allow temporal dynamics to interfere with the learning of spatial information. Such erroneous causal relationships lead to inaccurate spatial modeling, which in turn compromises generalizability. In contrast, STUNet explicitly models spatial information, thereby avoiding interference from temporal information and achieving more robust generalizability.

5.3 Performance Comparisons (RQ2)

Table 3 presents the results of selected strong baseline methods and our STUNet across four datasets (for full results of performance comparisons, please refer to 7 in appendix D). According to the results in Table 3, our STUNet has achieved performance comparable to existing state-of-the-art (SOTA) models. Furthermore, STUNet achieves the best performance in terms of the RMSE metric across all datasets and horizons, indicating that STUNet provides superior accuracy for larger values. This implies that STUNet is more precise in forecasting during peak periods, which typically require more intensive human resource scheduling and are more critical in urban governance. Consequently, STUNet holds greater practical significance for urban management.

5.4 Abliation Study (RQ3)

To address RQ2, we conducted ablation experiments on two datasets, SD and GBA. Specifically, we evaluated the impact of the following components on STUNet:

- "w/o spatial tokens": STUNet completely removes the spatial tokenizer and performs full attention only on temporal tokens.
- "w/o tokenizer pre-train": STUNet randomly initializes the spatial and temporal tokenizers and trains them directly alongside the entire model.
- "w/o tokenizer freeze": STUNet pre-trains the spatial and temporal tokenizers but does not freeze the spatial tokenizer during stage 2 training.
- "w/ full attention": STUNet replaces query-aggregate attention with full attention and employs absolute positional encoding for both temporal and spatial tokens.
- "w/o matrix augmentation": STUNet does not perform matrix permutation augmentation during the pre-training of the spatial tokenizer.
- "w/ reconstruction pre-train": STUNet utilizes a reconstruction task instead of a prediction task during the pre-training of the temporal tokenizer.

Based on the results shown in Table 4, we can draw the following conclusions:

Importance of spatial information in guiding temporal information fusion. According to the results of "w/o spatial tokens", removing spatial tokens led to the most severe performance degradation on both the SD and GBA datasets. Without spatial tokens, the model is forced to infer potential spatial relationships solely from the temporal information within the temporal tokens. This is causally flawed, and the experimental results further demonstrate that spatial information cannot be accurately reconstructed from temporal information alone. Additionally, "w/o tokenizer pre-train" and "w/o tokenizer freeze" also resulted in performance declines on both datasets. In these scenarios, spatial and temporal information are trained in a coupled manner, allowing temporal dynamics to interfere with spatial learning. Pre-training and freezing the spatial tokenizer decouples spatial feature extraction from temporal learning, thereby preventing such interference.

Table 1: Results of zero-shot performance comparison of our STUNet and baselines on traffic velocity datasets. Here redfont indicates the best performance and blurfont denotes the second best performance. All metrics are average values from horizon 1 to 12.

Methods	SD → GBA			SD → GLA			GBA → SD			GBA → GLA			GLA → SD			GLA → GBA		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
DLinear	47.82	75.17	42.49	51.71	81.86	44.74	50.06	81.02	40.20	51.81	81.36	46.84	50.45	82.32	36.80	48.23	75.94	42.32
PatchTST	48.54	77.93	40.27	50.61	83.64	30.72	50.13	83.28	31.47	49.22	81.68	30.55	50.58	82.93	31.99	47.29	74.96	39.88
STID	38.88	62.12	30.21	42.49	68.39	27.28	42.04	69.07	28.25	42.65	68.94	32.96	42.57	69.27	30.79	38.85	63.45	35.12
iTransformer	43.33	77.43	40.28	41.29	74.88	26.63	43.15	72.40	26.49	38.30	67.13	24.82	44.64	74.59	32.22	38.80	63.46	34.37
STGCN	38.58	60.37	34.88	43.17	69.97	37.71	34.62	52.20	26.40	35.57	53.56	28.53	33.46	51.74	23.94	32.88	49.81	27.60
STWave	45.56	69.46	46.42	50.80	76.53	66.37	48.33	73.56	37.27	53.85	82.62	48.67	49.01	77.19	38.65	45.49	69.67	40.43
PatchSTG	37.98	59.35	30.70	42.76	65.59	35.50	36.70	61.51	25.18	40.98	63.35	35.23	34.61	57.80	24.71	33.40	52.89	28.82
STUNet	34.46	53.90	26.88	36.77	57.98	24.55	32.71	52.16	21.90	33.06	50.96	24.42	32.19	49.44	22.41	32.39	47.52	26.41

Table 2: Results of zero-shot performance comparison of our STUNet and baselines on traffic flow datasets. Here redfont indicates the best performance and blurfont denotes the second best performance. All metrics are average values from horizon 1 to 12.

Methods	SD → GBA			SD → GLA			GBA → SD			GBA → GLA			GLA → SD			GLA → GBA		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
DLinear	47.82	75.17	42.49	51.71	81.86	44.74	50.06	81.02	40.20	51.81	81.36	46.84	50.45	82.32	36.80	48.23	75.94	42.32
PatchTST	48.54	77.93	40.27	50.61	83.64	30.72	50.13	83.28	31.47	49.22	81.68	30.55	50.58	82.93	31.99	47.29	74.96	39.88
STID	38.88	62.12	30.21	42.49	68.39	27.28	42.04	69.07	28.25	42.65	68.94	32.96	42.57	69.27	30.79	38.85	63.45	35.12
iTransformer	43.33	77.43	40.28	41.29	74.88	26.63	43.15	72.40	26.49	38.30	67.13	24.82	44.64	74.59	32.22	38.80	63.46	34.37
STGCN	38.58	60.37	34.88	43.17	69.97	37.71	34.62	52.20	26.40	35.57	53.56	28.53	33.46	51.74	23.94	32.88	49.81	27.60
STWave	45.56	69.46	46.42	50.80	76.53	66.37	48.33	73.56	37.27	53.85	82.62	48.67	49.01	77.19	38.65	45.49	69.67	40.43
PatchSTG	37.98	59.35	30.70	42.76	65.59	35.50	36.70	61.51	25.18	40.98	63.35	35.23	34.61	57.80	24.71	33.40	52.89	28.82
STUNet	34.46	53.90	26.88	36.77	57.98	24.55	32.71	52.16	21.90	33.06	50.96	24.42	32.19	49.44	22.41	32.39	47.52	26.41

Effectiveness of query-aggregate attention. The "w/ full attention" variant led to significant performance drops on both datasets. Since spatial and temporal tokens originate from two entirely independent tokenizers, they do not share the same semantic space. Furthermore, the positional information of temporal tokens is determined by their corresponding sensors and time steps, whereas that of spatial tokens depends on the relative positions of sensors; thus, their spatial positional information is also distinct. Full attention conflicts with these two facts, leading to substantial performance degradation. In contrast, the asymmetric processing of spatial and temporal tokens in query-aggregate attention enables effective information interaction while preserving their distinct semantic spaces.

Richness of the adjacency matrix. On the SD dataset, omitting matrix permutation augmentation during spatial tokenizer pre-training resulted in a significant performance decline, whereas only a minor drop was observed on the GBA dataset. Considering that the average node degree is nearly identical for both datasets, but the number of nodes in SD is approximately 30% of that in GBA, it implies that their traffic connectivity is similar. However, the diversity of adjacency matrices accessible in the SD dataset is much lower than in GBA. Consequently, the richness gain provided by matrix permutation augmentation is significantly more pronounced on the SD dataset.

Consistency between pre-training and downstream tasks. Replacing the pre-training task of the temporal tokenizer with reconstruction led to performance declines on both datasets. This indicates that employing the same task for temporal tokenizer pre-training can provide superior temporal tokens for stage 2.

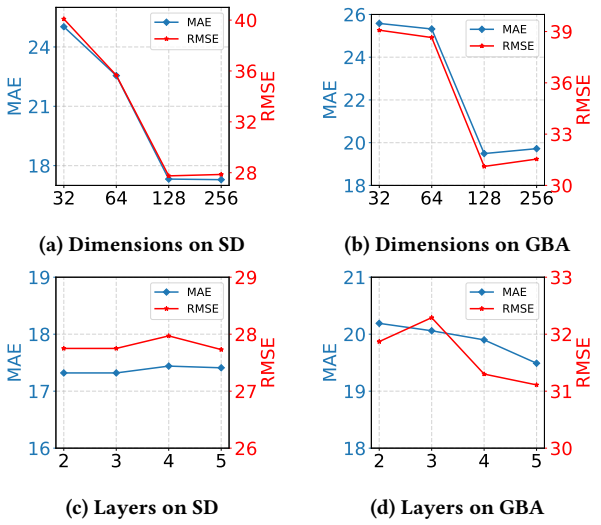


Figure 3: Hyper-parameter study of STUNet.

5.5 Hyper-parameters Study (RQ4)

Figure 3 and 4 illustrates the impact of several hyperparameters on model performance. We conducted experiments on the SD and GBA datasets, varying the hidden dimension across [32, 64, 128, 256] and the number of model layers across [2, 3, 4, 5]. Regarding dimension, we observed a distinct ceiling effect on both datasets: when the dimension is below 128, reducing it leads to severe performance degradation; however, increasing the dimension beyond 128 yields no further performance gains. This is because our tokens must accommodate spatial information with a patch_size of 64 and

Table 3: Traffic forecasting performance comparison of our STUNet and selected baselines. Redfont indicates the best performance and bluefont denotes the second best performance.

Datasets	Methods	Horizon 3			Horizon 6			Horizon 12			Average		
		MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
SD	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
	D^2STGNN	14.92	24.95	9.56	17.52	29.24	11.36	22.62	37.14	14.86	17.85	29.51	11.54
	DGCRN	15.34	25.35	10.01	18.05	30.06	11.90	22.06	37.51	15.27	18.02	30.09	12.07
	BigST	16.42	26.99	10.86	18.88	31.60	13.24	23.00	38.59	15.92	18.80	31.73	12.91
	STWave	15.80	25.89	10.34	18.18	30.03	11.96	21.98	36.99	15.30	18.22	30.12	12.20
	GMAN	16.20	27.32	11.78	19.64	33.59	13.55	25.71	44.73	16.83	20.01	34.97	13.81
	MAGE	15.09	25.22	10.10	18.86	31.18	15.29	25.30	41.07	20.10	18.72	31.59	14.00
	STID	15.15	25.29	9.82	17.95	30.39	11.93	21.82	38.63	15.09	17.86	31.00	11.94
	PatchSTG	14.53	24.34	9.22	16.86	28.63	11.11	20.66	36.27	14.72	16.90	29.27	11.23
	STUNet	14.87	23.73	9.55	17.38	27.55	11.47	21.18	33.21	14.69	17.32	27.75	11.62
GBA	DSTAGNN	19.73	31.39	15.42	24.21	37.70	20.99	30.12	46.40	28.16	23.82	37.29	20.16
	D^2STGNN	17.54	28.94	12.12	20.92	33.92	14.89	25.48	40.99	19.83	20.71	33.65	15.04
	DGCRN	18.02	29.49	14.13	21.08	34.03	16.94	25.25	40.63	21.15	20.91	33.83	16.88
	BigST	18.70	30.27	15.55	22.21	35.33	18.54	26.98	42.73	23.68	21.95	35.54	18.50
	STWave	17.95	29.42	13.01	20.99	34.01	15.62	24.96	40.31	20.08	20.81	33.77	15.76
	GMAN	18.10	30.09	14.71	23.01	38.07	18.32	31.54	52.49	25.45	23.44	41.06	18.80
	MAGE	18.18	28.61	13.17	23.23	35.46	19.66	28.62	43.15	25.62	22.79	35.40	19.68
	STID	17.36	29.39	13.28	20.45	34.51	16.03	24.38	41.33	19.90	20.22	34.61	15.91
	PatchSTG	16.81	28.71	12.25	19.68	33.09	14.51	23.49	39.23	18.93	19.50	33.16	14.64
	STUNet	16.86	27.10	12.38	19.68	31.11	15.13	23.44	36.37	19.83	19.49	31.11	15.25
GLA	DSTAGNN	19.49	31.08	11.50	24.27	38.43	15.24	30.92	48.52	20.45	24.13	38.15	15.07
	BigST	18.38	29.40	11.68	22.22	35.53	14.48	27.98	44.74	19.65	22.08	36.00	14.57
	STWave	17.48	28.05	10.06	21.08	33.58	12.56	25.82	41.28	16.51	20.96	33.48	12.70
	GMAN	17.41	29.53	18.76	22.02	37.72	22.20	29.78	51.23	28.16	22.47	39.45	22.68
	MAGE	16.86	26.58	12.11	21.64	32.20	16.72	27.92	40.37	26.02	21.51	32.44	18.22
	STID	16.54	27.73	10.00	19.98	34.23	12.38	24.29	42.50	16.02	19.76	34.56	12.41
	PatchSTG	15.84	26.34	9.27	19.06	31.85	11.30	23.32	39.64	14.60	18.96	32.33	11.44
	STUNet	16.48	26.05	10.35	19.75	31.08	12.41	24.33	37.94	16.09	19.64	31.34	12.53
CA	BigST	17.15	27.92	13.03	20.44	33.16	15.87	25.49	41.09	20.97	20.32	33.45	15.91
	STWave	16.77	26.98	12.20	18.97	30.69	14.40	25.36	38.77	19.01	19.69	31.58	14.58
	GMAN	17.54	26.81	14.50	20.38	32.30	15.30	27.07	43.94	17.70	21.13	34.13	15.68
	MAGE	17.03	25.89	16.25	21.46	31.85	22.25	26.87	38.87	31.62	21.00	31.77	21.95
	STID	15.51	26.23	11.26	18.53	31.56	13.82	22.63	39.37	17.59	18.41	32.00	13.82
	PatchSTG	14.69	24.82	10.51	17.41	29.43	12.83	21.20	36.13	16.00	17.35	29.79	12.79
	STUNet	14.87	24.15	10.83	17.75	28.36	13.36	21.87	35.05	17.11	17.65	28.22	13.23

Table 4: Ablation study of STUNet on average results of SD and GBA datasets. Bold indicates the best performance.

Dataset	SD			GBA			GLA		
Metric	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
w/o spatial tokens	20.85	33.05	13.71	23.97	35.67	18.06	24.28	37.69	15.40
w/o tokenizer pre-train	18.15	28.93	12.21	20.10	32.31	15.96	20.48	32.54	13.37
w/o tokenizer freeze	17.78	27.89	12.21	20.13	31.98	16.35	20.31	32.43	13.34
w/ full attention	19.52	31.33	13.28	20.88	32.46	17.78	23.54	36.10	15.83
w/o matrix augmentation	18.55	29.02	12.91	19.68	31.27	15.30	21.24	32.94	14.92
w/ reconstruction pre-train	19.43	31.13	13.47	20.28	32.60	16.50	21.10	32.71	14.09
STUNet	**17.32**	**27.75**	**11.62**	**19.49**	**31.11**	**15.25**	**19.64**	**31.34**	**12.53**

36-dimensional temporal information and facilitate their fusion. An insufficient dimension cannot adequately represent this information, resulting in suboptimal performance, whereas an excessively large dimension introduces redundancy. Regarding the number of layers, there is negligible performance difference between 2-layer and 3-layer networks on the SD dataset, and further increasing the

Table 5: Efficiency comparisons on four datasets. Here BS refers to batch size, Train refers to training time (in seconds) per epoch, Infer refers to inference time (in seconds) and Total refers to total training time (in hours). Note that Total denotes the whole training time. Besides, - indicates out of memory with minimum batch size 1.

Methods	SD				GBA				GLA				CA			
	BS	Train	Infer	Total	BS	Train	Infer	Total	BS	Train	Infer	Total	BS	Train	Infer	Total
DLinear	64	23	3	0.3	64	75	9	1	64	123	16	2	64	276	36	4
DCRNN	64	867	150	28	64	1,816	319	59	43	2,491	435	81	19	4,845	851	158
AGCRN	64	92	15	3	64	536	83	17	45	1,413	245	46	–	–	–	–
STGCN	64	53	16	2	64	160	54	6	64	268	86	10	64	701	206	25
GWNET	64	97	14	3	64	483	66	15	64	1,028	139	32	44	4,105	548	113
ASTGCN	64	128	19	4	45	1,126	147	35	17	3,060	393	77	–	–	–	–
STGODE	64	188	26	6	49	710	103	23	30	1,305	192	42	13	4,212	659	135
DSTAGNN	64	240	23	7	27	1,959	171	53	10	5,241	467	120	–	–	–	–
DGCRN	64	430	76	14	12	4,461	605	138	–	–	–	–	–	–	–	–
D^2STGNN	45	563	69	14	4	5,885	796	148	–	–	–	–	–	–	–	–
STID	64	34	5	0.5	64	111	16	2	64	181	27	3	64	408	60	6
PatchSTG	64	64	6	1	64	262	30	4	64	295	27	4	32	981	93	14
STUNet	64	150	46	4	64	680	180	8	64	1140	312	12	32	2417	594	22

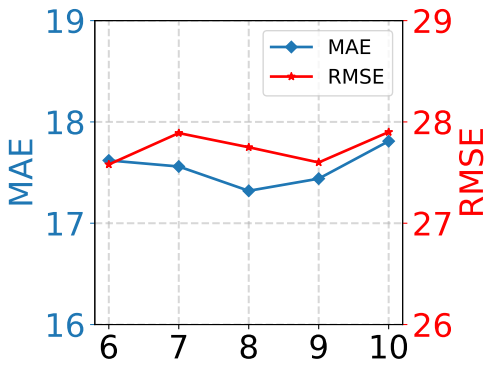


Figure 4: Patch size on SD

depth may even lead to performance decline. In contrast, on the GBA dataset, increasing the number of layers from 2 to 5 generally results in performance improvements. This suggests that for larger datasets, the model requires deeper representations to capture the increased diversity in the data. Besides, the result of figure 4 indicates that STUNet's performance is not sensitive to the patch size of spatial tokenizer with enough dimension.

5.6 Efficiency Comparisons (RQ5)

To evaluate the efficiency of our proposed STUNet, we compared its training and inference speeds with those of other spatio-temporal models. As shown in Table 5, STUNet demonstrates high efficiency among spatio-temporal models. As the number of nodes in the dataset increases, several GNN-based methods encounter Out-of-Memory (OOM) issues or exhibit quadratic complexity growth, whereas the increase in training and inference time for STUNet is significantly more gradual. This efficiency is attributed to our patching operation on the adjacency matrix, which reduces the complexity of modeling spatial relationships. This becomes increasingly pronounced as the number of nodes grows.

6 Conclusion

In this paper, we introduce the Spatio-Temporal Unified Network (STUNet), a framework designed to explicitly model spatial and temporal features into unified representations. By employing a spatial tokenizer that partitions the relational graph into patches, STUNet successfully extracts robust spatial tokens that remain invariant to temporal fluctuations, thereby ensuring strong cross-network generalizability. Furthermore, our proposed query-aggregate attention mechanism effectively integrates these disparate representations by simulating the causal relationships between upstream and downstream nodes. Extensive experimental results on multiple traffic forecasting benchmarks demonstrate that STUNet not only achieves competitive performance but also exhibits superior zero-shot generalizability and computational efficiency. We believe that the explicit modeling paradigm and the decoupled training strategy presented in this work provide a promising direction for building scalable and robust intelligent transportation systems.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62322606, No. 62441605).

References

[1] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. In *NeurIPS*.

[2] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Conguri Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. 2020. Spectral temporal graph neural network for multivariate time-series forecasting. In *NeurIPS*.

[3] Srinivasa Ravi Chandra and Haitham Al-Deek. 2009. Predictions of freeway traffic speeds and volumes using vector autoregressive models. *Journal of Intelligent Transportation Systems* (2009).

[4] Dexiong Chen, Leslie O'Bray, and Karsten Borgwardt. 2022. Structure-Aware Transformer for Graph Representation Learning. In *ICML*.

[5] Zheng Dong, Renhe Jiang, Haotian Gao, Hangchen Liu, Jinliang Deng, Qingsong Wen, and Xuan Song. 2024. Heterogeneity-Informed Meta-Parameter Learning for Spatiotemporal Time Series Forecasting. In *Proceedings of SIGKDD*.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

[7] Taoran Fang, Tianhong Gao, Chunping Wang, Yihao Shang, Wei Chow, Lei Chen, and Yang Yang. 2025. KAA: Kolmogorov-Arnold Attention for Enhancing Attentive Graph Neural Networks. In *ICLR*.

[8] Yuchen Fang, Yuxuan Liang, Bo Hui, Zezhi Shao, Liwei Deng, Xu Liu, Xinke Jiang, and Kai Zheng. 2025. Efficient Large-Scale Traffic Forecasting with Transformers: A Spatial Data Management Perspective. In *Proceedings of SIGKDD*.

[9] Yuchen Fang, Yanjun Qin, Haiyong Luo, Fang Zhao, Bingbing Xu, Liang Zeng, and Chenxing Wang. 2023. When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks. In *ICDE*.

[10] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. 2021. Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting. In *Proceedings of SIGKDD*.

[11] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *AAAI*.

[12] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*.

[13] Jindong Han, Weijia Zhang, Hao Liu, Tao Tao, Naiqiang Tan, and Hui Xiong. 2024. BigST: Linear Complexity Spatio-Temporal Graph Neural Network for Traffic Forecasting on Large-Scale Road Networks. *Proceedings of the VLDB Endowment* (2024).

[14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked Autoencoders Are Scalable Vision Learners. In *CVPR*.

[15] Renhong Huang, Yuxuan Cao, Yi Li, Junwei Hu, Zihua Xiong, Shuai Fang, Sheng Guo, Bo Zheng, and Yang Yang. 2026. Trimming the Fat: Redundancy-Aware Acceleration Framework for DGNNs. In *AAAI*, Vol. 40. 22003–22011.

[16] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. 2023. PDFormer: Propagation Delay-aware Dynamic Long-range Transformer for Traffic Flow Prediction. In *AAAI*.

[17] Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. 2023. Spatio-temporal meta-graph learning for traffic forecasting. In *AAAI*.

[18] Weiwei Jiang and Jiayun Luo. 2022. Graph neural network for traffic forecasting: A survey. *Expert Syst. Appl.* (2022).

[19] Guangyin Jin, Yuxuan Liang, Yuchen Fang, Zezhi Shao, Jincai Huang, Junbo Zhang, and Yu Zheng. 2024. Spatio-Temporal Graph Neural Networks for Predictive Learning in Urban Computing: A Survey. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[20] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[21] S Vasantha Kumar and Lelitha Vanajakshi. 2015. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *European Transport Research Review* (2015).

[22] Hyunwook Lee and Sungahn Ko. 2024. TESTAM: A Time-Enhanced Spatio-Temporal Attention Model with Mixture of Experts. In *ICLR*.

[23] Mengzhang Li and Zhanxing Zhu. 2021. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *AAAI*.

[24] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR*.

[25] Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. 2023. LargeST: A Benchmark Dataset for Large-Scale Traffic Forecasting. In *NeurIPS*.

[26] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *ICLR*.

[27] Zhongjian Lv, Jiajie Xu, Kai Zheng, Hongzhi Yin, Pengpeng Zhao, and Xiaofang Zhou. 2018. Lc-rnn: A deep learning model for traffic speed prediction.. In *Proceedings of IJCAI*.

[28] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A time series is worth 64 words: Long-term forecasting with transformers. *ICLR* (2023).

[29] Cheonbok Park, Chunggi Lee, Hyojin Bahng, Yunwon Tae, Seungmin Jin, Kihwan Kim, Sungahn Ko, and Jaegul Choo. 2020. ST-GRAT: A Novel Spatio-temporal Graph Attention Networks for Accurately Forecasting Dynamically Changing Road Speed. In *Proceedings of CIKM*.

[30] Xiangyu Zhao Haoliang Li Hongwei Zhao Yiqi Wang Zitao Liu Qian Ma, Zijian Zhang and Wanyu Wang. 2023. Rethinking Sensors Modeling:Hierarchical Information Enhanced Traffic Forecasting. In *Proceedings of CIKM*.

[31] Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. 2022. Recipe for a General, Powerful, Scalable Graph Transformer. In *NeurIPS*.

[32] Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. 2022. Spatial-Temporal Identity: A Simple yet Effective Baseline for Multivariate Time Series Forecasting. In *Proceedings of CIKM*.

[33] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomput.* (2024).

[34] Yifei Sun, Yang Yang, Xiao Feng, Zijun Wang, Haoyang Zhong, Chunping Wang, and Lei Chen. 2025. Handling Feature Heterogeneity with Learnable Graph Patches. In *Proceedings of SIGKDD*.

[35] Shihao Tu, Yang Yang, Wenyue Ding, Yicheng Lu, Qingkai Ren, Yupeng Zhang, and Yin Zhang. 2025. ASTNet: Asynchronous Spatio-Temporal Network for Large-Scale Chemical Sensor Forecasting. In *Proceedings of SIGKDD*.

[36] Shihao Tu, Yupeng Zhang, Jing Zhang, Zhendong Fu, Yin Zhang, and Yang Yang. 2024. Powerpm: Foundation model for power systems. *NeurIPS* 37 (2024), 115233–115260.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NeurIPS*.

[38] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *ICLR* (2018).

[39] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. 2022. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381* (2022).

[40] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *NeurIPS* (2021).

[41] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of SIGKDD*.

[42] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of IJCAI*.

[43] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *ICLR*.

[44] Zhongchao Yi, Zhengyang Zhou, Qihe Huang, Yanjiang Chen, Liheng Yu, Xu Wang, and Yang Wang. 2024. Get Rid of Isolation: A Continuous Multi-task Spatio-Temporal Learning Framework. In *NeurIPS*.

[45] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do Transformers Really Perform Badly for Graph Representation?. In *NeurIPS*.

[46] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of IJCAI*.

[47] Haitao Yuan and Guoliang Li. 2021. A Survey of Traffic Prediction: from Spatio-Temporal Data to Intelligent Transportation. *Data Science and Engineering* (2021).

[48] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting?. In *AAAI*.

[49] Yunhao Zhang and Junchi Yan. 2023. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *ICLR*.

[50] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems* (2019). doi:10.1109/TITS.2019.2935152

[51] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. GMAN: A Graph Multi-Attention Network for Traffic Prediction. In *AAAI*.

[52] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*.

[53] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML*.

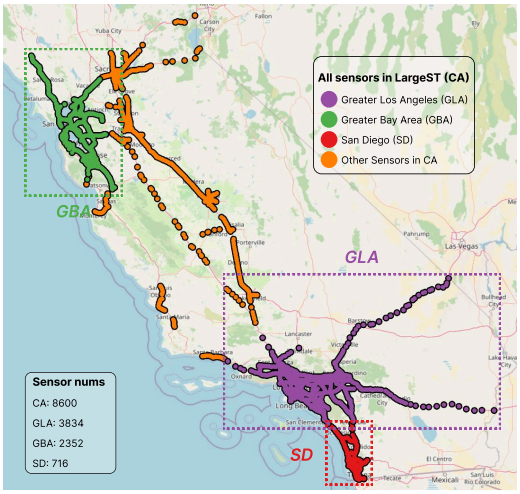


Figure 5: Geometric Distribution of Sensors in LargeST

A Dataset Statistics

We utilized four datasets from LargeST: SD, GBA, GLA, and CA. LargeST contains five years of traffic flow data (from 2017 to 2021) with a 5-minute interval. For our experiments, we used the data from 2019 and downsampled it to a 15-minute interval, resulting in 35,040 time steps for each sensor, which are widely accepted. Table 6 presents the statistical data for these four datasets. Although the number of sensors varies significantly across the datasets, their average degrees are remarkably similar, suggesting a degree of consistency in their spatial dependencies. We further visualized the distribution of the number of first-order neighbors for the sensors in each dataset, as shown in Figure 6. It can be observed that the degree distributions across the four datasets are quite similar, all exhibiting a unimodal structure with most degrees concentrated between 10 and 40. Slight variations exist among the datasets; for instance, the peak for the GBA dataset is around 30, while the peaks for the other three datasets are around 20. Additionally, the distribution of high-degree sensors in the SD dataset is relatively uniform, whereas in the other three datasets, the number of high-degree sensors decreases significantly as the degree increases. Furthermore, we visualized the geographical distribution of sensors across the different datasets, as illustrated in Figure 5. It is evident that there is no overlap between the SD, GBA, and GLA datasets, which provides a solid foundation for our zero-shot experiments.

Table 6: Dataset statistics.

Datasets	Sensors	Edges	Degree	Time Range
SD	716	17,319	24.2	01/01/2019-12/31/2019
GBA	2,352	61,246	26.0	01/01/2019-12/31/2019
GLA	3,834	98,703	25.7	01/01/2019-12/31/2019
METR-LA	207	1515	7.32	03/01/2012-06/30/2012
PEMS-BAY	325	2369	7.29	01/01/2017-05/31/2017
SZ-TAXI	156	532	3.41	01/01/201501/31/2015

B Implementation Details

To help better reproduce our experimental results, here are the implementation details of STUNet.

B.1 Data Sampling Strategy

All datasets from LargeST are too large to be directly processed. Therefore, we employ a data sampling strategy to convert the whole datasets into batches. To be specific, we partition all sensors of the dataset into groups with batch size B_s. A batch contains a group of sensors and their one-hop neighbors with multiple series of batch size B_t. Such sampling strategy can maintain spatial relations between sensors and reduce memory usage. Besides, sensor-grouping follows a random strategy rather than a distance-based strategy(such as Kmeans). In other words, sensors in a batch may not have correlations. This will force STUNet to distinguish relative and non-relative sensors and extract features from relative sensors, thus enhancing robustness of STUNet.

B.2 Training Statistics

STUNet employs a two-stage training strategy: tokenizer pre-training and backbnone training. During tokenizer pre-training, both spatial tokenizer and temporal tokenizer are trained for 20 epochs on SD and GBA, 10 epochs on GLA and CA. Besides, we conduct a random permutation for each adjacency matrix in batches to enhance the diversity of adjacency matrix during spatial tokenizer pre-training, thus making spatial tokenizer robust. We employ ReduceLROnPlateau as learning rate scheduler with initial learning rate 0.0005, factor 0.3 and patience 10 for all datasets. During backbone training, we train for 120, 100, 90 and 80 epochs on SD, GBA, GLA and CA. We adopt ReduceLROnPlateau lr scheduler with initial learning rate 0.001 and patience 10 for all datasets, and the factors of SD, GBA, GLA and CA are 0.5, 0.3, 0.2 and 0.1. In addition, both tokenizer pre-training and backbone training are optimized with Adam. The whole framework is implemented with pytorch and run on one NVIDIA H800 80GB GPU.

C Analysis of distributions of spatial tokens

Real-world traffic networks are composed of various prototypical topological structures. To evaluate whether the spatial tokenizer of STUNet can effectively discriminate between these patterns, we sampled instances of straight roads, Y-intersections, and cycles from the SD dataset. We then extracted their corresponding spatial tokens using the pre-trained spatial tokenizer. Following an average pooling operation on these tokens, we employed UMAP for clustering visualization and quantitative assessment. The experimental results yielded an Adjusted Rand Index (ARI) of 0.96 and a Normalized Mutual Information (NMI) of 0.98, indicating that the spatial tokenizer successfully recognizes distinct road structures. Furthermore, a Silhouette Score of 0.62 was achieved; values exceeding 0.5 typically signify robust cluster separation. These findings demonstrate that STUNet's spatial tokenizer is capable of distinguishing diverse spatial relationships, thereby providing strong empirical evidence for the rationality of our architectural design.

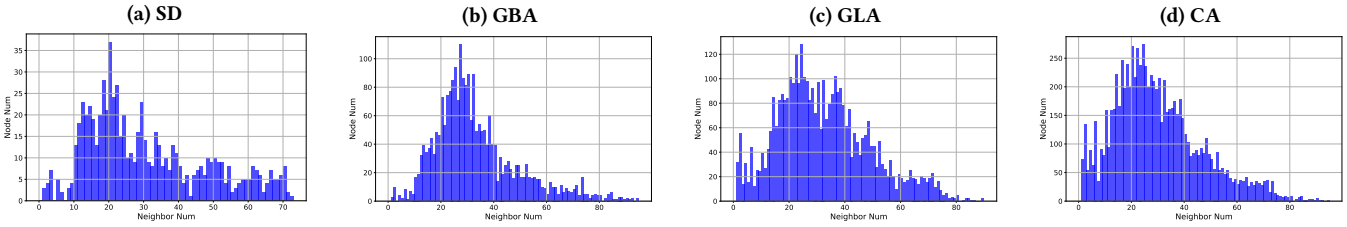


Figure 6: Sensor degree distribution of the four datasets

Table 7: Extra traffic forecasting results in addition to Table 3

Datasets	Methods	Horizon 3			Horizon 6			Horizon 12			Average		
		MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
SD	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
	GWNET	15.24	25.13	9.86	17.74	29.51	11.70	21.56	36.82	15.13	17.74	29.62	11.88
	AGCRN	15.71	27.85	11.48	18.06	31.51	13.06	21.86	39.44	16.52	18.09	32.01	13.28
	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
GBA	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
	GWNET	17.85	29.12	13.92	21.11	33.69	17.79	25.58	40.19	23.48	20.91	33.41	17.66
	AGCRN	18.31	30.24	14.27	21.27	34.72	16.89	24.85	40.18	20.80	21.01	34.25	16.90
	STGODE	18.84	30.51	15.43	22.04	35.61	18.42	26.22	42.90	22.83	21.79	35.37	18.26
GLA	DCRNN	18.41	29.23	10.94	23.16	36.15	14.14	30.26	46.85	19.68	23.17	36.19	14.40
	STGCN	19.86	34.10	12.40	22.75	38.91	14.11	26.70	45.78	17.00	22.64	38.81	14.17
	GWNET	17.28	27.68	10.18	21.31	33.70	13.02	26.99	42.51	17.64	21.20	33.58	13.18
	AGCRN	17.27	29.70	10.78	20.38	34.82	12.70	24.59	42.59	16.03	20.25	34.84	12.87
	STGODE	18.10	30.02	11.18	21.71	36.46	13.64	26.45	45.09	17.60	21.49	36.14	13.72
CA	DCRNN	17.55	28.21	12.68	21.79	34.27	16.67	28.56	44.34	23.84	21.87	34.41	17.06
	STGCN	18.99	32.37	14.84	21.37	36.46	16.27	24.94	42.59	19.74	21.33	36.39	16.53
	GWNET	17.14	27.81	12.62	21.68	34.16	17.14	28.58	44.13	24.24	21.72	34.20	17.40
	STGODE	17.57	29.91	13.91	20.98	36.62	16.88	25.46	45.99	21.00	20.77	36.60	16.80

D Additional Results

We present the additional results of performance comparison experiment here in Table 7.