ST-LLM+: Graph Enhanced Spatio-Temporal Large Language Models for Traffic Prediction

Chenxi Liu, Kethmi Hirushini Hettige, Qianxiong Xu, Cheng Long, Shili Xiang, Gao Cong, Ziyue Li, Rui Zhao

***Abstract*—Traffic prediction is a crucial component of data management systems, leveraging historical data to learn spatio-temporal dynamics for forecasting future traffic and enabling efficient decision-making and resource allocation. Despite efforts to develop increasingly complex architectures, existing traffic prediction models often struggle to generalize across diverse datasets and contexts, limiting their adaptability in real-world applications. In contrast to existing traffic prediction models, large language models (LLMs) progress mainly through parameter expansion and extensive pre-training while maintaining their fundamental structures. In this paper, we propose ST-LLM+, the graph enhanced spatio-temporal large language models for traffic prediction. Through incorporating a proximity-based adjacency matrix derived from the traffic network into the calibrated LLMs, ST-LLM+ captures complex spatio-temporal dependencies within the traffic network. The Partially Frozen Graph Attention (PFGA) module is designed to retain global dependencies learned during LLMs pre-training while modeling localized dependencies specific to the traffic domain. To reduce computational overhead, ST-LLM+ adopts the LoRA-augmented training strategy, allowing attention layers to be fine-tuned with fewer learnable parameters. Comprehensive experiments on real-world traffic datasets demonstrate that ST-LLM+ outperforms state-of-the-art models. In particular, ST-LLM+ also exhibits robust performance in both few-shot and zero-shot prediction scenarios. Additionally, our case study demonstrates that ST-LLM+ captures global and localized dependencies between stations, verifying its effectiveness for traffic prediction tasks.**

***Index Terms*—Traffic Prediction, Large Language Models, Spatio-Temporal Data**

I. Introduction

Rapid urban expansion coupled with evolving mobility demands has resulted in several challenges, including traffic congestion, urban sprawl, and increased vulnerability to climate change impacts [1]–[3]. Data-driven frameworks have emerged to address these issues by improving traffic flow, optimizing route planning, and enhancing overall safety [4]–[6]. Traffic prediction, which aims to predict future traffic features like traffic flow at specific locations using historical data, is a crucial component of data management systems [7]–[9]. This predictive capability is essential for optimizing traffic management [10], [11] and scheduling public transportation [12]. For example, accurately predicting bike flow benefits the transportation department in optimizing bike management. Similarly, forecasting taxi flow is vital for taxi companies, as it enables them to efficiently allocate and schedule vehicles to satisfy expected demand [13]–[15].

The evolution of traffic prediction has seen a shift from traditional statistical methods to deep learning techniques [16], [17]. Initially, statistical methods such as the Autoregressive Integrated Moving Average (ARIMA) and Kalman Filter were adapted to fit the time series data. In addition, machine learning techniques, including Logistic Regression, Support Vector Regression, K-Nearest Neighbors, and Random Forest have been utilized to tackle the problem of traffic prediction. However, these models are not good at capturing spatio-temporal dependencies within traffic data, leading to the rapid development of deep learning solutions.

The deep learning techniques for traffic prediction are further divided into small classic models and large models based on the parameters in the models. The classic models often adopt, linear layer [18], convolutional neural networks (CNNs) [19], graph neural networks (GNNs) [20], and transformers [21], [22] to learn spatial and temporal dependencies and spatial locations and time steps for future forecasting. However, public benchmark datasets are often limited in scale and diversity [23], constraining the use of deeper layers (i.e., models with more parameters). This limitation results in shallow feature extraction, reducing the potential effectiveness of these classic models. Recently, large language models (LLMs) have surfaced as dominant paradigms achieving state-of-the-art performances in fields such as Computer vision [24] and Natural Language Processing [25]. Inspired by the unique strengths of foundation models, such as the ability to leverage large-scale parameters, enhanced generalizability, and inherent robustness in handling diverse scenarios, the utilization of LLMs has emerged as a robust alternative in time series [26], [27] and spatio-temporal analysis [28], [29].

While originating from different domains, the sequential nature of both spatio-temporal data and natural language makes it reasonable to adapt LLMs for forecasting tasks. Spatio-temporal data, similar to language, is characterized by its sequential dependencies, where prior events influence future outcomes. This similarity makes LLMs particularly effective for such tasks since their extensive pre-training on diverse textual datasets has sharpened their capability to detect complex patterns and dependencies within large

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contributions from the industry partner(s). *(Chenxi Liu and Kethmi Hirushini Hettige contribute equally to this work.)*

Chenxi Liu, Qianxiong Xu, and Cheng Long are with the S-Lab, Nanyang Technological University, Singapore. (e-mail: {chenxi.liu, qianxiong.xu, c.long}@ntu.edu.sg). Kethmi Hirushini Hettige and Gao Cong are with the College of Computing and Data Science, Nanyang Technological University, Singapore. (e-mail: kethmihi001@e.ntu.edu.sg and gaocong@ntu.edu.sg). Shili Xiang is with the Institute for Infocomm Research, A$*$STAR, Singapore. (e-mail: sxiang@i2r.a-star.edu.sg). Ziyue Li is with the Information System Department, University of Cologne, 50923, Germany. (e-mail: zlibn@connect.ust.hk). Rui Zhao is with SenseTime Research, Beijing 100190, China. (e-mail: zhaorui@sensetime.com).

data sets. Recognizing this, several unified frameworks have been proposed to leverage the unique capabilities of large language models (LLMs) for time series and spatio-temporal data across diverse applications. These frameworks capitalize on LLMs' pre-training on large datasets, enabling them to generalize effectively and adapt to new tasks. Parameter-tuning approaches modify LLMs to suit time series tasks [30], [31], while other methods preprocess time series data into formats like textual prompts or use reprogramming to fit the LLMs input space [32], [33]. Beyond adaptation, some techniques involve developing foundation models specifically designed for temporal data analysis [34].

The majority of existing LLM-based prediction methods predominantly focus on the temporal dimension of data in traffic prediction tasks [26], [27], [30] and often overlook the spatial dependencies inherent in traffic networks. However, traffic prediction is not purely a temporal task, as congestion often tends to move from one road link to its adjacent streets, where a station's future conditions are influenced by its past and the activities of nearby stations [35]. In our preliminary study [28], we addressed this limitation by treating the time steps of a spatial location as a token and modeling the global temporal dependencies across these spatial tokens. While this approach captures some spatial dependencies by aggregating temporal tokens across locations, it fails to represent the complex and non-sequential spatial dependencies arising from proximity-based interactions, which cannot be effectively modeled through simple tokenization. For example, the traffic patterns at two distant but connected locations may exhibit dependencies that are overlooked when spatial dependencies are modeled sequentially. Additionally, the masked multi-head self-attention (MMSA) mechanism employed in LLMs like GPT-2 limits the model's ability to capture spatial dependencies by forcing each token (i.e., spatial location) to attend only to its previous tokens in the sequence. This prevents the model from considering spatial locations ahead of the sequence that may have strong dependencies with the current location. This restricts the model's ability to fully capture the complex spatial interactions across different locations, leading to suboptimal performance in tasks where spatial dependencies are crucial.

To address the aforementioned limitations, we initially proposed partially frozen attention (PFA) LLMs specifically designed to enhance traffic prediction accuracy. By partially freezing the multi-head attention layers, the LLMs can adapt to traffic prediction while preserving the foundational knowledge acquired during pre-training. This strategy ensures that the extensive pre-trained capabilities to detect and comprehend complex patterns in sequential data are retained and can be directly utilized for analyzing sequential traffic data. Advancing this further, our extended model incorporates graph-based information through a customized attention mask to capture more intricate structural spatial dependencies. By introducing the adjacency matrix into the attention mechanism within the foundation model, we allow the model to learn the spatial dependencies directly from the graph structure, enabling it to attend to spatial locations that are not sequentially aligned but are highly correlated within the traffic network. The graph-based attention focuses on local spatial dependencies, which is crucial for accurate traffic prediction. This advancement addresses the inability of prior models to fully capture the spatial dimension and narrows the domain gap between language-based models and traffic data.

In summary, we propose ST-LLM+, an enhanced Spatio-Temporal Large Language Model for traffic prediction. In this new ST-LLM+ framework, we embody graph-based attention to capture complex spatio-temporal dependencies that cannot be represented through simple tokenization and node embeddings. This model integrates the adjacency matrix of spatial locations into the attention mechanism, allowing nodes to attend to their neighboring locations directly and capturing their structural dependencies. While the node embedding process remains consistent with our preliminary work, we extend the partially frozen attention strategy to handle spatial dependencies using graph structures, that is, partially frozen graph attention (PFGA). We also strengthen the model further by customizing the LoRA (Low-Rank Adaptation)-augmented [36] training strategy to efficiently fine-tune attention layers, improving the model's adaptability. Extensive experiments on real-world traffic datasets demonstrate the effectiveness of ST-LLM+, showing its improvements over existing methods. The main contributions of this paper are outlined below:

- We propose graph enhanced spatio-temporal large language models for traffic prediction, which introduces graph-based attention within calibrated LLMs to capture complex spatio-temporal dependencies within the traffic network.
- We extend the partially frozen attention strategy to PFGA. The global dependencies are captured using the standard multi-head attention mechanism, while local spatial dependencies are captured by introducing the proximity-based adjacency matrix as an attention mask in the last layers of the calibrated LLMs.
- We design the LoRA-augmented training strategy for the attention layers of LLMs for efficient fine-tuning of the model with smaller computational overhead.
- Extensive experiments on real traffic datasets show the superior performance of ST-LLM+ across various settings, improving upon previous approaches. Furthermore, the few-shot and zero-shot prediction capabilities demonstrate the model's robustness in both intra-domain and inter-domain knowledge transfer.

In comparison to the ST-LLM from our preliminary work [28], ST-LLM+ offers the following key improvements:

- **Enhanced Spatio-Temporal Dependency Modeling via Graph Attention within Calibrated LLMs**: ST-LLM+ integrates a novel graph attention mechanism by incorporating a proximity-based adjacency matrix to calibrate the LLMs, empowering the model to capture the intricate spatio-temporal dependencies inherent in traffic networks.
- **Partially Frozen Graph Attention Module**: ST-LLM+ extends the partially frozen attention to the PFGA by incorporating the adjacency matrices as attention masks in the unfrozen layers of the LLMs. This module allows ST-LLM+ to maintain the global dependencies learned in its pre-training while effectively capturing localized

dependencies specific to the traffic domain.

- **LoRA-augmented Fine-Tuning Strategy**: ST-LLM+ customizes the LoRA to fine-tune the attention layers of LLMs efficiently for reducing computational time while maintaining adaptability to diverse traffic scenarios.
- **Comprehensive Experiments and Discussions**: Extensive evaluations on traffic datasets demonstrate the superior performance of ST-LLM+ compared to its predecessor in terms of effectiveness and efficiency. The few-shot and zero-shot results highlight its robustness and transferability across tasks. Additionally, the case study demonstrates that ST-LLM+ captures global and localized dependencies between stations.

The remainder of this paper is as follows. Section II discusses related work about LLMs for spatio-temporal data and traffic prediction. Section III introduces the problem definition. Section IV details the ST-LLM+ model, followed by the experiments in Section V. Section VI concludes the paper.

II. Related Work

In this section, we review the related work from large language models for spatio-temporal data and traffic prediction.

A. Large Language Models for Spatio-Temporal Data

Initially, pre-trained LLMs mainly focusing on time series analysis tasks, such as forecasting [26], classification [27], anomaly detection [37], imputation [38], few-shot learning and zero-shot learning [33]. These methods leverage pre-trained knowledge and the transfer learning capabilities of LLMs, making them effective at contextualizing temporal dependencies. For instance, OFA [30], TEMPO-GPT [26], and LLM4TS [39] achieved successful time series forecasting performance across diverse datasets by fine-tuning the generative pre-trained transformer (GPT-2) backbone. In contrast to fine-tuning approaches, models like LLMTime leveraged LLMs for time series forecasting by representing numeric in text format and generating extrapolations through text completions exhibiting impressive zero-shot performance [33]. Moreover, there exist approaches that directly utilize different prompting techniques for time series tasks. For example, PromptCast [32] converts numerical time series data into prompts and uses them in LLMs directly to derive the predictions.

More recently, LLM-based techniques have increasingly utilized multiple data modalities [40], [41]. For example, Time-LLM reprogrammed an LLM for time series forecasting, and the backbone language model remained intact [27]. Similarly, METS [42] incorporates a trainable encoder and a frozen LLM to process paired ECG data and clinical reports. Correspondingly, UniTime introduces a versatile model that leverages text instructions and a Language-TS Transformer to align domain-specific characteristics for effective forecasting across multiple time series domains [31]. To align the time series and textual modalities, S^2IP-LLM [43] integrates the pre-trained semantic space with time series embeddings space and performs time series forecasting based on learned prompts from the joint space. Existing time series forecasting methods focus solely on the temporal dimension of the data and fail to capture the complex spatial dependencies for accurate forecasting, especially in applications like traffic prediction.

In contrast to LLMs for standard time series, research on LLM-based spatio-temporal models is still in its early stages. Our preliminary work, ST-LLM [28], is one of the leading studies at the forefront of this domain. ST-LLM redefines the time series of each location as tokens, integrating a spatio-temporal embedding module along with a novel partially frozen attention strategy to effectively capture spatio-temporal dependencies for traffic prediction. Additionally, there exist UrbanGPT [29] and UniST [44] frameworks that utilize LLMs as enhancers by either augmenting the spatio-temporal data or improving the existing modeling framework with enriched external knowledge derived from LLMs. Specifically, UrbanGPT enables LLMs to act as a bridge, enriching spatio-temporal models with external knowledge by combining textual instructions with spatio-temporal data through an instruction-tuning paradigm [29]. UniST is a versatile model that uses LLMs to enhance spatio-temporal analysis by generating prompts that align spatio-temporal patterns from diverse urban datasets, improving generalization and prediction performance across various tasks like crowd flow modeling [44]. Despite their advancements, these studies cannot explore the capabilities of LLMs for modeling graph structure, which is vital for accurate traffic prediction.

B. Traffic Prediction

Traffic prediction aims to predict future traffic features based on historical traffic data, which is a crucial component in intelligent transportation systems [45], [46]. Traffic data is a special type of time series data. Thus, it is natural to adapt the classical time series models, such as ARIMA, VAR, and Kalman filter, for the traffic prediction tasks in the early stage . Kumar et al. used the seasonal ARIMA model for short-term traffic prediction [47], while Lu et al. utilized VAR [48] as an extension of the autoregressive (AR) model to account for linear inter-dependencies among multiple time series. Chang et al. proposed a tensor-extended Kalman filter framework to characterize nonlinear dynamics and applied it to traffic forecasting [49]. These models, however, do not perform effectively on traffic data because of the inherent complex spatio-temporal dependencies involved.

Later, numerous efforts have been dedicated to advancing traffic prediction techniques by developing various neural network-based models. In the beginning, CNNs were applied to traffic data to capture spatial dependencies in the data. Shen et al. divided the city into grids and applied 3D CNNs for traffic prediction [50]. Since CNNs are primarily designed to be applied in regular, grid-like urban areas, they encounter challenges when dealing with the non-Euclidean spatial structure of traffic data. This irregularity makes it difficult for CNNs to accurately capture the spatial dependencies inherent in traffic data. The swift progress in graph learning has made graph convolutional network (GCN)-based models popular due to their permutation-invariance, local connectivity, and compositional nature [51]–[53]. Li et al. modeled the traffic data as a directed graph and introduced a diffusion convolutional recurrent network (DCRNN), which integrates diffusion convolution with gated recurrent unit (GRU) layers, enabling it to capture spatio-temporal dependencies [54]. Subsequently, several extensions to this model were proposed, incorporating meta-learning with graph attention networks [55] and node-specific adaptive parameters in graph convolution [53] to further improve performance. Choi et al. presented a graph neural controlled differential equation for traffic prediction [56]. Temporal Convolutional Network (TCN)-based models like STGCN [57] and GWNet [58] adopted dilated causal convolutions for temporal modeling, leading to faster training times and strong performance on various benchmarks. However, GCN-based models suffer from over-smoothing, making it hard to capture global dependencies [56].

Recently, attention-based models have emerged as a dominant trend [59]–[62]. Without taking the adjacency matrix into account, attention-based models can still model dynamic spatial dependencies effectively. Initial studies in this domain, including ASTGCN [59] and STAEformer [63] demonstrate superior ability in capturing long-range dependencies and handling diverse traffic patterns. In [61], the authors developed an attention-based spatio-temporal graph neural network for traffic prediction. However, the structures of these models are becoming increasingly sophisticated.

III. Preliminaries

In this section, we introduce the preliminaries of the traffic prediction, with key notations summarized in Table I.

Definition 1. **Traffic Data:** We denote the traffic data as $\mathbf{X} \in \mathbb{R}^{T \times N \times C}$, where T is the number of time steps, N is the number of spatial stations, and C is the traffic feature. For example, $C = 1$ represents the traffic pick-up or drop-off flow.

Definition 2. **Traffic Network:** We formulate traffic network as a graph $G = (V, E, A)$, where V is a set of $|V| = N$ nodes, each corresponding to a traffic station. $E \subseteq V \times V$ represents the set of edges, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix representing the proximity between nodes.

Definition 3. **Traffic Prediction:** Given the historical traffic data of P time steps $\mathbf{X}_P = \{\mathbf{x}_{t-P+1}, \mathbf{x}_{t-P+2}, \ldots, \mathbf{x}_t\} \in \mathbb{R}^{P \times N \times C}$ and traffic graph G, the objective is to learn a function $f(\cdot)$ with parameter θ to predict traffic data of on the following S time steps $\mathbf{Y}_S = \{\mathbf{y}_{t+1}, \mathbf{y}_{t+2}, \ldots, \mathbf{y}_{t+S}\} \in \mathbb{R}^{S \times N \times C}$:-

$$[\mathbf{x}_{t-P+1}, \mathbf{x}_{t-P+2}, \ldots, \mathbf{x}_t, G] \xrightarrow[\theta]{f(\cdot)} [\mathbf{y}_{t+1}, \mathbf{y}_{t+2}, \ldots, \mathbf{y}_{t+S}]. \tag{1}$$

IV. Methodology

In this section, we provide a detailed elaboration of the proposed ST-LLM+ and its components.

A. Overview

The proposed ST-LLM+ framework is illustrated in Figure 1, which integrates a spatio-temporal embedding layer, a fusion convolution layer, Partially Frozen Graph Attention (PFGA) LLMs, and a regression convolution layer. The historical traffic data is initially denoted as $\mathbf{X}_P$, which contains N tokens of spatial locations. The $\mathbf{X}_P$ is processed through the spatio-temporal embedding layer, which extracts the token embedding of historical P time steps, spatial embedding, and temporal embedding, as $\mathbf{E}_T \in \mathbb{R}^{N \times D}$, $\mathbf{E}_S \in \mathbb{R}^{N \times D}$, and $\mathbf{E}_P \in \mathbb{R}^{N \times D}$, respectively. A fusion convolution then integrates these representations into a unified way $\mathbf{E}_F \in \mathbb{R}^{N \times 3D}$. Subsequently, the $\mathbf{E}_F$ and the adjacency matrix $\mathbf{A}$ representing the spatial dependencies between nodes is input into the LoRA-augmented PFGA LLMs.

TABLE I: Summary of Notations

Notation	Description
$\mathbf{X}$	Traffic Data
N	Number of spatial locations in $\mathbf{X}$
T	Number of time steps in $\mathbf{X}$
C	Number of features in $\mathbf{X}$
P	Number of historical time steps used in traffic prediction
S	Number of future time steps for traffic prediction
$\mathbf{X}_P$	Historical Traffic Data
$\mathbf{Y}_S$	Future Traffic Data
G	Traffic Graph
V	Nodes or spatial locations of G
E	Edges of G
$\mathbf{A}$	Proximity-based adjacency matrix of G
$\mathbf{E}_P$	Token embedding of historical P time steps
$\mathbf{E}_T$	Temporal embedding
$\mathbf{E}_S$	Spatial embedding
$\mathbf{H}^L$	Final output of the PFGA LLMs Module

The PFGA LLMs encompass $F+U$ layers, where the multi-head attention and feed-forward layers in the first F layers are frozen to preserve the pre-trained knowledge while the multi-head attention layers in the last U layers are unfrozen and replaced with graph-based attention to enhance the model's focus on capturing the spatio-temporal dependencies between tokens. Furthermore, LoRA (Low-Rank Adaptation) is applied to the last U attention layers to reduce the number of trainable parameters. The resulting output of the PFGA LLMs module is represented as $\mathbf{H}^L \in \mathbb{R}^{N \times 3D}$. Finally, the regression convolution layer takes H^L and predicts the following traffic data, denoted as $\widehat{\mathbf{Y}}_S \in \mathbb{R}^{S \times N \times C}$.

B. Spatio-Temporal Embedding and Fusion

We fist define the time steps at each location of traffic data as tokens. The spatio-temporal embedding layer transforms the tokens into spatio-temporal representations that align with the LLMs. These representations include spatial dependencies, hour-of-day, day-of-week patterns, and token information.

We embed the tokens through a pointwise convolution, where the input data $\mathbf{X}_P$ is transformed into the embedding $\mathbf{E}_P \in \mathbb{R}^{N \times D}$:

$$\mathbf{E}_P = PConv(\mathbf{X}_P; \theta_p), \tag{2}$$

where $\mathbf{E}_P$ represents the token embedding. $PConv$ denotes the point-wise convolution operation using filters with a 1×1 kernel size. $\mathbf{X}_P$ is the input traffic data, D is the hidden dimension. θ_p represents the learnable parameters of the pointwise convolution.

To preserve temporal information within the tokens, we utilize linear layers to encode traffic data into separate embeddings for hour-of-day and day-of-week temporal embeddings. We perform absolute positional encoding for each traffic data

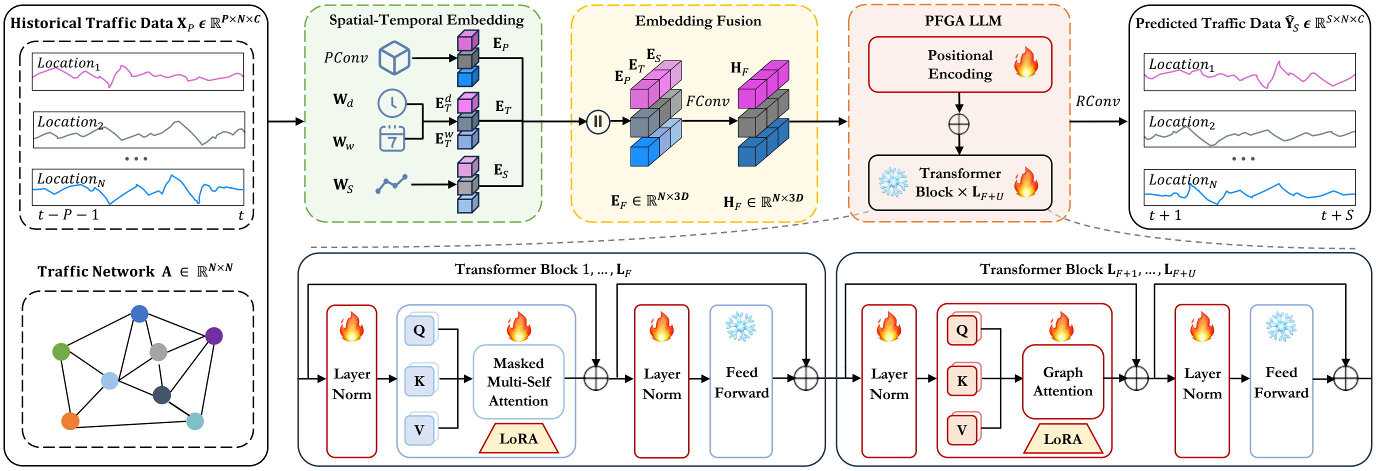

Historical Traffic Data $\mathbf{X}_P \in \mathbb{R}^{P\times N\times C}$
$Location_1$
$Location_2$
$Location_N$
$t-P-1$
t
Traffic Network $\mathbf{A} \in \mathbb{R}^{N\times N}$
Spatial-Temporal Embedding
$PConv$
$\mathbf{W}_d$
$\mathbf{W}_w$
$\mathbf{W}_S$
$\mathbf{E}_P$
$\mathbf{E}_T^d$
$\mathbf{E}_T^w$
$\mathbf{E}_T$
$\mathbf{E}_S$
Embedding Fusion
$\mathbf{E}_P$
$\mathbf{E}_T$
$\mathbf{E}_S$
$FConv$
$\mathbf{H}_F$
$\mathbf{E}_F \in \mathbb{R}^{N\times 3D}$
$\mathbf{H}_F \in \mathbb{R}^{N\times 3D}$
PFGA LLM
Positional Encoding
Transformer Block × $\mathbf{L}_{F+U}$
$RConv$
Predicted Traffic Data $\hat{\mathbf{Y}}_S \in \mathbb{R}^{S\times N\times C}$
$Location_1$
$Location_2$
$Location_N$
$t+1$
$t+S$
Transformer Block 1, ..., $\mathbf{L}_F$
Layer Norm
Q
K
V
Masked Multi-Self Attention
LoRA
Layer Norm
Feed Forward
Transformer Block $\mathbf{L}_{F+1}$, ..., $\mathbf{L}_{F+U}$
Layer Norm
Q
K
V
Graph Attention
LoRA
Layer Norm
Feed Forward

Fig. 1: ST-LLM+ Framework. ST-LLM+ modeling traffic data via a **Spatio-Temporal Embedding**. These embeddings are integrated uniformly by an **Embedding Fusion** layer. The **LoRA-augmented Partially Frozen Graph Attention (PFGA) LLMs** consist of $F + U$ layers: the first F layers employ the original multi-head attention mechanism, while the last U layers incorporate graph-based attention using an adjacency matrix as the attention mask. The multi-head attention and feed-forward layers in the first F layers are frozen, while multi-head attention layers are unfrozen and enhanced with LoRA in the last U layers. Finally, the output from the PFGA LLMs is passed through a regression layer to generate the traffic prediction results.

at the "day" and "week" resolutions, and the generated positional encodings are $\mathbf{X}_{day} \in \mathbb{R}^{N\times T_d}$ and $\mathbf{X}_{week} \in \mathbb{R}^{N\times T_w}$. The hour-of-day embedding $\mathbf{E}_T^d \in \mathbb{R}^{N\times D}$ and day-of-week embedding $\mathbf{E}_T^w \in \mathbb{R}^{N\times D}$ are calculated as follows:

$$\mathbf{E}_T^d = \mathbf{W}_d(\mathbf{X}_{day}), \tag{3}$$

$$\mathbf{E}_T^w = \mathbf{W}_w(\mathbf{X}_{week}), \tag{4}$$

$$\mathbf{E}_T = \mathbf{E}_T^d + \mathbf{E}_T^w, \tag{5}$$

where $\mathbf{W}_d \in \mathbb{R}^{T_d\times D}$ and $\mathbf{W}_w \in \mathbb{R}^{T_w\times D}$ are the learnable parameter embeddings for the hour-of-day and day-of-week, respectively. By adding these two embeddings, we obtain the temporal representation $\mathbf{E}_T \in \mathbb{R}^{N\times D}$.

To represent spatial dependencies among token pairs, we design an adaptive embedding $\mathbf{E}_S \in \mathbb{R}^{N\times D}$:

$$\mathbf{E}_S = \sigma(\mathbf{W}_S \cdot \mathbf{X}_P + \mathbf{b}_s) \tag{6}$$

where σ denotes the activation function. $\mathbf{W}_S$ and $\mathbf{b}_s$ are the learnable parameter.

Subsequently, we introduce a fusion convolution $FConv$ to project the traffic data to the required dimensions of the LLMs. Specifically, the $FConv$ integrates the token, spatial, and temporal embeddings to represent each token uniformly:

$$\mathbf{H}_F = FConv(\mathbf{E}_P||\mathbf{E}_S||\mathbf{E}_T;\theta_f) \tag{7}$$

where $\mathbf{H}_F \in \mathbb{R}^{N\times 3D}$. $||$ denotes concatenation and θ_f represents the learnable parameters of the $FConv$.

C. Partially Frozen Graph Attention Module

In our framework, we integrate graph-based attention into partially frozen LLMs, referred to as PFGA LLMs. While the frozen pre-trained transformer (FPT) has demonstrated effectiveness in various downstream tasks across non-language modalities [64], they often fall short when dealing with tasks like traffic prediction, which requires capturing both short-term and long-term dependencies. To address this, we introduce a partially frozen GPT-2 architecture, enhanced with graph-based attention, leveraging proximity-based adjacency matrices to model spatial dependencies effectively.

Similar to our original ST-LLM, we freeze the first F layers of GPT-2. Accordingly, ST-LLM+ uses the adjacency matrix as the attention mask within the final U unfrozen layers. This allows our model to capture local spatial dependencies in a more structured way by enabling each spatial location to attend to previous nodes in the sequence and other neighboring spatial locations based on their proximity in the graph structure.

In the first F layers of the PFGA LLMs, we freeze the multi-head attention and feed-forward layers:

$$\begin{aligned}\bar{\mathbf{H}}^i &= MHA\left(LN\left(\mathbf{H}^i\right)\right) + \mathbf{H}^i,\\ \mathbf{H}^{i+1} &= FFN\left(LN\left(\bar{\mathbf{H}}^i\right)\right) + \bar{\mathbf{H}}^i,\end{aligned} \tag{8}$$

where the range of i is from 1 to $F-1$, and $\mathbf{H}^1 = [\mathbf{H}_F + PE]$. PE represents the learnable positional encoding. $\bar{\mathbf{H}}^i$ represents the intermediate representation of the i_{th} layer after applying the frozen multi-head attention $MHA(\cdot)$ and the first unfrozen layer normalization $LN(\cdot)$. $\mathbf{H}^i$ symbolizes the final representation after applying the unfrozen LN and frozen feed-forward network $FFN(\cdot)$. These layers are defined as follows:

$$\begin{aligned}&LN\left(\mathbf{H}^i\right) = \gamma \odot \frac{\mathbf{H}^i - \mu}{\sigma} + \beta,\\ &MHA(\tilde{\mathbf{H}}^i) = \mathbf{W}^O(head_1||\cdots||head_h),\\ &head_i = Attention(\mathbf{W}_i^Q\tilde{\mathbf{H}}^i, \mathbf{W}_i^K\tilde{\mathbf{H}}^i, \mathbf{W}_i^V\tilde{\mathbf{H}}^i),\\ &Attention(\tilde{\mathbf{H}}^i) = \text{softmax}\left(\frac{\tilde{\mathbf{H}}^i\tilde{\mathbf{H}}^{iT}}{\sqrt{d_k}}\right)\tilde{\mathbf{H}}^i,\\ &FFN(\hat{\mathbf{H}}^i) = \max(0, \mathbf{W}_1\hat{\mathbf{H}}_P^{i+1} + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2,\end{aligned} \tag{9}$$

where $\tilde{\mathbf{H}}^i$ is the output of $\mathbf{H}^i$ after passing through the first $LN(\cdot)$. $\hat{\mathbf{H}}^i$ is the output of $\bar{\mathbf{H}}^i$ after the second $LN(\cdot)$. γ and β are learnable scaling and translation parameters. μ and σ represent the mean and standard deviation, respectively. $\odot$ denotes element-wise multiplication.

In the last U layers of the LLM, we unfreeze the MHA and integrate the graph adjacency matrix $\mathbf{A}$ into the attention mechanism to adapt the ST-LLM+ for capturing spatio-temporal dependencies of traffic data. The attention is calibrated as follows:

$$\begin{aligned}\bar{\mathbf{H}}^{\mathbf{F+U-1}} &= MHA\left(LN\left(\mathbf{H}^{\mathbf{F+U-1}}\right), A\right) + \mathbf{H}^{\mathbf{F+U-1}},\\ \mathbf{H}^{\mathbf{F+U}} &= FFN\left(LN\left(\bar{\mathbf{H}}^{\mathbf{F+U-1}}\right)\right) + \bar{\mathbf{H}}^{\mathbf{F+U-1}},\end{aligned} \tag{10}$$

where $\bar{\mathbf{H}}^{F+U}$ represents the intermediate representation of the L_{F+U-1} layer after applying the unfrozen MHA and the second frozen LN. Here, the adjacency matrix A serves as an attention mask, restricting attention to neighboring nodes based on the graph structure. $\mathbf{H}^{F+U}$ denotes the final output of the $\mathbf{L}_{F+U}$ layer after applying both the unfrozen LN and frozen FFN, with the MHA being unfrozen.

D. LoRA-augmented Training Strategy

We incorporate LoRA, which allows us to fine-tune the model's attention mechanism using low-rank matrices, significantly reducing the number of trainable parameters while maintaining flexibility in the attention layers. In particular, LoRA is integrated into the attention layers of GPT-2, as illustrated in Figure 1. This approach enhances the model's ability to adapt to the graph structure without overfitting or requiring extensive computational resources.

We apply LoRA [36] to the attention layers of the PFGA LLMs to improve scalability and efficiency. LoRA is a fine-tuning technique designed to significantly reduce the number of trainable parameters in large models by introducing low-rank matrices that approximate the updates to key layers. In our context, we use LoRA to introduce low-rank matrices $\mathbf{W}_q$ and $\mathbf{W}_c$ into the query and context attention mechanisms, allowing the model to adapt its attention with fewer parameters. In the original attention mechanism as shown in Equation (8), the query (Q), key (K), and value (V) matrices are computed using $\mathbf{H}$ as the hidden state input, and $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, and $\mathbf{W}_i^V$ as the weight matrices respectively.

However, fine-tuning these full matrices would involve updating a large number of parameters, which becomes computationally expensive in tasks involving large spatial networks. Hence, we use LoRA to reduce this complexity by decomposing the updates to the query and value matrices into low-rank approximations. Specifically, LoRA introduces trainable low-rank matrices $\mathbf{L}_i^Q \in \mathbb{R}^{3D\times r}$ and $\mathbf{M}_i^Q \in \mathbb{R}^{r\times d_k}$, with r being the rank of the approximation ($r \ll d_k$) as follows:

$$\Delta\mathbf{W}_i^Q = \mathbf{L}_i^Q\mathbf{M}_i^Q, \quad \Delta\mathbf{W}_i^V = \mathbf{L}_i^V\mathbf{M}_i^V, \tag{11}$$

$$\mathbf{W'}_i^Q = \mathbf{W}_i^Q + \Delta\mathbf{W}_i^Q, \quad \mathbf{W'}_i^V = \mathbf{W}_i^V + \Delta\mathbf{W}_i^V. \tag{12}$$

Algorithm 1: The ST-LLM+ Framework

```
Input: Traffic data $\mathbf{X}_P$ in the historical $P$ time steps,
       adjacency matrix $\mathbf{A}$ and all hyperparameters.
Output: Trained ST-LLM.
for each epoch do
    Shuffle training data
    for each batch $\mathbf{X}_P$ in training data do
        $\mathbf{E}_F \leftarrow$ Spatio-Temporal Embedding by
          Equations (2), (3), (4) and (6) with $\mathbf{X}_P$.
        $\mathbf{H}_F \leftarrow$ Embedding Fusion by Equation (7)
          with $\mathbf{E}_F$.
        for $i = 1$ to $F + U$ do
            $\mathbf{H}^1 \leftarrow$ PFA LLMs Initialization with $\mathbf{H}_F$.
            if $i \leq F$ then
                calculate $\mathbf{H}^{i+1}$ by Equation (8) with
                  $\mathbf{H}^i$.
            else
                calculate $\mathbf{H}^{F+U}$ by Equation (10) with
                  $\mathbf{H}^i$.
            end
        end
        $\widehat{\mathbf{Y}}_S \leftarrow$ by Equation (14).
        Update all learnable parameters by minimizing
          the loss in Equation (15) with $\widehat{\mathbf{Y}}_S$ and $\mathbf{Y}_S$
          via Ranger21 optimizer.
    end
end
```

With LoRA applied, the attention is updated as follows:

$$\begin{aligned}&MHA(\tilde{\mathbf{H}}^i) = \mathbf{W}^O(head_1^{\text{lora}}||\cdots||head_h^{\text{lora}}),\\ &head_i^{\text{lora}} = Attention(\mathbf{W}_i^{Q'}\tilde{\mathbf{H}}^i, \mathbf{W}_i^{K'}\tilde{\mathbf{H}}^i, \mathbf{W}_i^{V'}\tilde{\mathbf{H}}^i).\end{aligned} \tag{13}$$

E. Traffic Prediction

Regression convolution $RConv(\cdot)$ predicts the traffic data on the following S time steps:

$$\hat{\mathbf{Y}}_S = RConv(\mathbf{H}^{F+U}; \theta_r), \tag{14}$$

where $\widehat{\mathbf{Y}}_S \in \mathbb{R}^{S\times N\times C}$ and θ_r represents the learnable parameters of the regression convolution.

The loss function of ST-LLM+ is established as follows:

$$\mathcal{L} = \left\|\widehat{\mathbf{Y}}_S - \mathbf{Y}_S\right\| + \lambda \cdot L_{\text{reg}}, \tag{15}$$

where $\widehat{\mathbf{Y}}_S$ is the predicted traffic data, $\mathbf{Y}_S$ is the ground truth, L_{reg} represents the L2 regularization term, and λ is a hyperparameter. The whole process of the ST-LLM+ is shown in Algorithm 1, while Algorithm 2 shows the detailed process of the LoRA-augmented PFGA LLMs.

F. Complexity Analysis

We analyze the time complexity of the proposed methods. In the ST-LLM+, the fusion operation for embeddings across nodes involves token, spatial, and temporal embeddings with a complexity of $O(Vd)$, where V is the number of nodes

Algorithm 2: LoRA-augmented PFGA LLMs

```
Input: Input hidden state $\mathbf{H}$, adjacency matrix $\mathbf{A}$,
       pre-trained GPT-2 layers, LoRA rank $r$, and all
       hyperparameters.
Output: $\mathbf{H}^{F+U} \leftarrow$ Final output of PFGA LLMs
for each layer $i = 1$ to $F + U$ do
    if $i \leq F$ then
        Freeze the attention and feed-forward layers.
        Calculate $\mathbf{H}^{i+1} \leftarrow$ Frozen Attention and FFN
          by Equation (8) with $\mathbf{H}^i$.
    else
        Compute $\Delta\mathbf{W}_i^Q$ and $\Delta\mathbf{W}_i^V$ by Equation (11).
        Compute $\mathbf{W}'^Q_i$ and $\mathbf{W}'^V_i$ by Equation (12).
        Use adjacency matrix $\mathbf{A}$ as the attention mask
          (Equation (8).
        Compute attention $MHA(\tilde{\mathbf{H}}^i)$ by
          Equation (10).
    end
end
```

and d is the hidden dimension. The time complexity of the PFGA LLMs consists of frozen and unfrozen layers: the first frozen F layers, the complexity is $O(FV^2d)$, while the last unfrozen U layers, which incorporate graph-based attention, have a complexity of $O(UEd)$, where E represents the number of edges as determined by the adjacency matrix. Summing these components, the overall complexity of ST-LLM+ per forward pass is $O((FV^2 + U(E+V))d)$. For our previous ST-LLM, the time complexity of PFA LLMs is also split into frozen and unfrozen layers. In the first F frozen layers, the model performs standard multi-head self-attention on the node embeddings. The time complexity is $O(FV^2d)$. In the last U unfrozen layers, the model applies trainable attention layers with a time complexity of $O(UV^2d)$. Thus, the total time complexity per forward pass for ST-LLM is $O((F+U)V^2d)$. In summary, as E is usually far smaller than V on the traffic datasets used in this paper because of the sparse adjacency matrices, we can approximate $E+V < V^2$. Given the example on the CHBike dataset, we have $E = 9,798$ and $V = 250$. Therefore, we find $E + V = 9,798 + 250 = 10,048$ and $V^2 = 250 \times 250 = 62,500$. Since $E + V = 10,048 < V^2 = 62,500$, the time complexity of ST-LLM+ is lower than that of our previous ST-LLM.

V. Experiments

In this section, we aim to validate the superiority of our ST-LLM+ through a series of extensive experimental evaluations.

A. Datasets

This section details the datasets employed to examine the predictive performance of the ST-LLM+ and baselines, with real-world traffic data from NYCTaxi[1] and CHBike[2].

TABLE II: Dataset Description.

Dataset Description	NYCTaxi	CHBike
Total Trips	35 million	2.6 million
Number of Stations	266	250
Number of Timesteps	4,368	4,368
Timestep Interval	30 minutes	30 minutes
Time Span	01/04/2016-30/06/2016	01/04/2016-30/06/2016

NYCTaxi. NYCTaxi dataset comprises over 35 million taxi trips in New York City (NYC), systematically categorized into 266 virtual stations. Spanning three months from April 1st to June 30th, 2016, it includes 4,368 time steps, each representing a half-hour interval.

CHBike. Consisting of approximately 2.6 million Citi bike orders, the CHBike dataset reflects the usage of the bike-sharing system in the same period as the NYCTaxi dataset, from April 1st to June 30th, 2016. After filtering out stations with few orders, it focuses on the 250 most frequented stations.

B. Baselines

We compare ST-LLM+ with the following baselines classified into three categories: (1) GNN-based models: DCRNN [54], STGCN [57], GWN [52], AGCRN [53], STSGCN [51], STG-NCDE [56], DGCRN [65]. (2) Attention-based models: ASTGCN [59], GMAN [60], ASTGNN [61]. (3) LLMs: OFA [30], GATGPT [38], LLaMA-2 [66], GCNGPT and ST-LLM [28]. The details of the baselines are outlined as follows:

- DCRNN [54]: An approach that models the data as a directed graph and introduces diffusion convolutional recurrent network.
- STGCN [57]: A graph convolutional network that combines 1D convolution to tackle the time series prediction task in the traffic domain.
- GWN [52]: A graph neural network that employs graph convolution with an adaptive adjacency matrix.
- AGCRN [53]: An adaptive graph convolutional recurrent network that incorporates node learning and interdependency inference among traffic series.
- STSGCN [51]: A modified graph convolutional network that captures localized spatio-temporal dependencies and heterogeneities by using synchronized modeling and modular components.
- DGCRN [65]: An approach that introduces a traffic prediction framework using dynamic graph convolutional recurrent networks.
- ASTGCN [59]: An attention-based spatio-temporal graph convolutional network for traffic forecasting.
- GMAN [60]: An attention-based predictive model that adopts an encoder-decoder architecture.
- ASTGNN [61]: An attention-based model for learning the dynamics and heterogeneity of traffic data.
- OFA [30]: An approach that modifies GPT-2 by freezing the self-attention and feed-forward networks within its residual blocks. We take an inverted view of the traffic data of OFA for better prediction performance.

[1] https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[2] https://citibikenyc.com/system-data

- GATGPT [38]: A model that incorporates a GAT before the GPT-2 backbone FPT [64]. We also implement GATGPT, where the GAT is after the GPT-2.
- GCNGPT [28]: A model that combines the GCN with the FPT GPT-2.
- LLaMA-2 [66]: A collection of pre-trained and fine-tuned large language models developed by Meta. In the LLaMA-2, we adapt the frozen pre-trained transformer.
- ST-LLM [28]: Our preliminary work introduces spatio-temporal large language models with a partially frozen attention mechanism.

C. Implementations

Aligning with contemporary practices, we divided the NYCTaxi and CHBike datasets into training, validation, and test sets using a 6:2:2 ratio. We set the historical time steps P and the future time steps S to 12 each. T_w is set at 7 to represent a week's seven days. T_d is 48, with each timestep spanning 30 minutes. The experiments were carried out on a system incorporating NVIDIA A100 GPUs, each with 40GB of memory. For training LLM-based models, we used the Ranger21 optimizer with a learning rate of 0.001, while GCN and attention-based models employed the Adam optimizer, also set at a 0.001 learning rate. The LLMs used are GPT-2 and LLaMA-2 7B. We configured GPT-2 with six layers [30], and LLaMA-2 with eight layers [27]. The batch size is 64, and the max training epoch is 500. Note that the experimental results are averaged across all prediction time steps. Code and datasets are publicly available[3].

D. Evaluation Metrics

Three metrics were used for evaluating the models: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Weighted Absolute Percentage Error (WAPE). MAE and RMSE quantify absolute errors, while WAPE assesses relative errors. In all metrics, lower values indicate superior prediction performance:

$$\text{MAE} = \frac{1}{m}\sum_{i=1}^{m}\left|\widehat{\mathbf{Y}}_i - \mathbf{Y}_i\right|, \tag{16}$$

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\widehat{\mathbf{Y}}_i - \mathbf{Y}_i\right)^2}, \tag{17}$$

$$\text{WAPE} = \frac{\sum_{i=1}^{m}\left|\widehat{\mathbf{Y}}_i - \mathbf{Y}_i\right|}{\sum_{i=1}^{m}\left|\mathbf{Y}_i\right|} \times 100\%, \tag{18}$$

where m is the number of predicted values.

E. Main Results

Table III displays the performance comparison results of ST-LLM+ with baseline models. The LLM used in ST-LLM + is GPT-2. We can make the following observations. (1) ST-LLM+ demonstrates the most effective performance across all evaluated metrics in both the NYCTaxi and CHBike datasets. The LoRA-augmented Partially Frozen Graph-based Attention (PFGA) and the use of adjacency matrices to capture spatial dependencies are key contributors to ST-LLM+'s superior performance. (2) Although OFA and LLaMA-2 demonstrate reasonable performance, ST-LLM+ significantly outperforms both, achieving a 7.7% improvement in MAE over OFA and 6.8% over LLaMA-2 on average across all datasets. This may be due to OFA's ineffective traffic data embedding and LLaMA-2's lack of spatio-temporal optimization. (3) For graph-based LLMs, GATGPT, GPTGAT, and GCNGPT fall short of fully capturing spatio-temporal dependencies due to their ineffective temporal representations. GPTGAT slightly outperforms GATGPT, which suggests that incorporating graph attention after the LLM backbone leads to better spatial feature learning than the reverse design in GATGPT. (4) Attention-based models like ASTGNN and GMAN perform variably, generally lagging behind ST-LLM+ due to the limitations of traditional attention in handling complex spatio-temporal embeddings. (5) GNN-based models, such as GWN and DGCRN, effectively capture spatial dependencies but have difficulty with long-range temporal dependencies, resulting in lower accuracy than ST-LLM+. In summary, the results highlight a clear trend in performance among different types of models, with LLM-based models leading in traffic prediction, followed by attention-based models and GCN-based models, underscoring the advanced capabilities of LLM-based approaches.

F. Ablation Studies

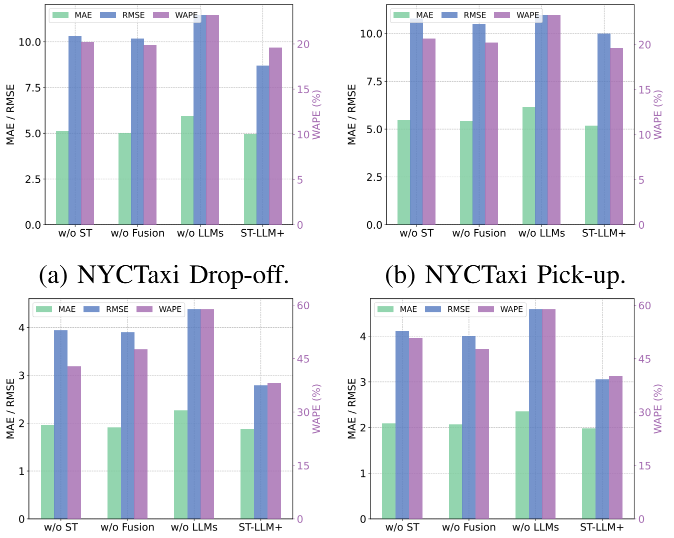
Ablation Study of ST-LLM+ Components: ST-LLM+ is composed of several key components, each contributing uniquely to its overall performance in traffic prediction tasks. To assess the impact of each component, we created variants, w/o ST: without the spatio-temporal embedding layer, w/o Fusion: without the fusion convolution layer, and w/o LLMs: without the LLMs component. Figure 2 presents the corresponding ablation study on the NYCTaxi and CHBike datasets. Removing the LLMs component (w/o LLMs) significantly increases errors, demonstrating that the prediction capabilities of the ST-LLM+ are heavily reliant on the LLM's ability to learn complex dependencies from traffic data. The exclusion of the spatio-temporal embedding layer (w/o ST) results in a notable performance drop, emphasizing its importance in modeling spatio-temporal dependencies. Furthermore, removing the fusion layer (w/o Fusion) also declines performance to a lesser extent than removing the LLMs or spatio-temporal embeddings as it effectively integrates spatial and temporal data for cohesive LLMs processing. The complete ST-LLM+ model achieves the lowest errors, underscoring the importance of combining the LLMs, spatio-temporal embedding, and fusion convolution for effective traffic prediction.

Ablation Study of PFGA LLMs: Next, we conducted an ablation study to evaluate the efficacy of our proposed PFGA LLMs by examining the impact of each design choice through different variants; Full Tuning (FT): Fully tuned LLMs without frozen layers, evaluating the benefit of selective freezing in PFGA, Full Graph-based Attention (FGA): Incorporates the

[3] https://github.com/ChenxiLiu-HNU/ST-LLM+

TABLE III: Model comparison on traffic datasets in terms of MAE, RMSE, and WAPE (%). Results are averaged from all prediction time steps. The bold and underlined font shows the best and the second-best results, respectively.

Models	NYCTaxi Pick-up			NYCTaxi Drop-off			CHBike Pick-up			CHBike Drop-off		
	MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE
DCRNN	5.40	9.71	20.43%	5.19	9.63	19.82%	2.09	3.30	42.26%	1.96	2.94	39.61%
STGCN	5.71	10.22	21.62%	5.38	9.60	20.55%	2.08	3.31	42.08%	2.01	3.07	40.62%
ASTGCN	7.43	13.84	28.04%	6.98	14.70	26.60%	2.76	4.45	55.71%	2.79	4.20	56.49%
GWN	5.43	9.39	20.55%	5.03	8.78	19.21%	2.04	3.20	40.95%	1.95	2.98	39.43%
AGCRN	5.79	10.11	21.93%	5.45	9.56	20.81%	2.16	3.46	43.69%	2.06	3.19	41.78%
GMAN	5.43	9.47	20.42%	5.09	8.95	19.33%	2.20	3.35	44.06%	2.09	3.00	42.00%
STSGCN	6.19	11.14	25.37%	5.62	10.21	22.59%	2.36	3.73	50.09%	2.73	4.50	54.10%
ASTGNN	5.90	10.71	22.32%	6.28	12.00	23.97%	2.37	3.67	47.81%	2.24	3.35	45.27%
STG-NCDE	6.24	11.25	23.46%	5.38	9.74	21.37%	2.15	3.97	61.38%	2.28	3.42	46.06%
DGCRN	5.44	9.82	20.58%	5.14	9.39	19.64%	2.06	3.21	41.51%	1.96	2.93	39.70%
OFA	5.82	10.42	22.00%	5.60	10.14	21.36%	2.06	3.21	41.70%	1.96	2.97	39.68%
GATGPT	5.92	10.55	22.39%	5.66	10.39	21.60%	2.07	3.23	41.70%	1.95	2.94	39.43%
GPTGAT	5.47	9.74	20.77%	5.58	10.23	20.85%	2.05	3.20	41.69%	1.94	2.92	39.37%
GCNGPT	6.58	12.23	24.88%	6.64	12.24	25.32%	2.37	3.80	47.66%	2.24	3.48	45.37%
LLaMA-2	5.35	9.48	20.27%	5.66	10.74	21.63%	2.10	3.37	42.49%	1.99	3.03	40.28%
ST-LLM	5.29	9.42	20.03%	5.07	9.07	19.18%	1.99	3.08	40.19%	1.89	2.81	38.27%
ST-LLM+	**5.18**	**8.98**	**19.60%**	**4.94**	**8.68**	**18.86%**	**1.98**	**3.05**	**40.11%**	**1.88**	**2.79**	**38.20%**



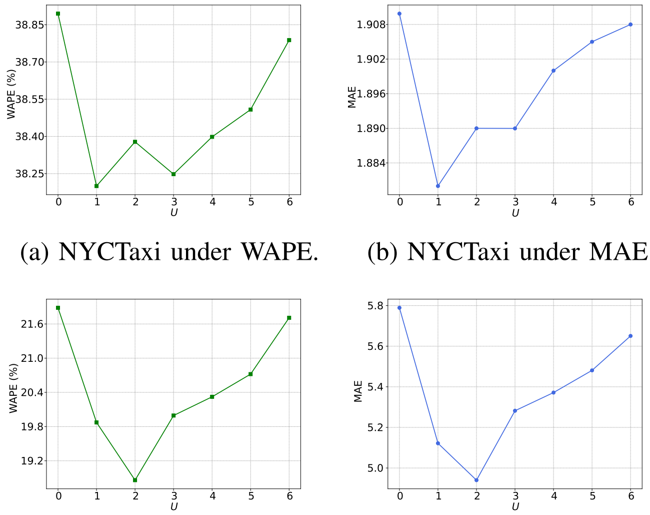
(a) NYCTaxi Drop-off.

(b) NYCTaxi Pick-up.

(c) CHBike Drop-off.

(d) CHBike Pick-up.

Fig. 2: Ablation Study of ST-LLM+ Key Components.



(a) NYCTaxi under WAPE.

(b) NYCTaxi under MAE.

(c) CHBike under WAPE.

(d) CHBike under MAE.

Fig. 3: Performance Study of Unfreezing Last U Layers on Drop-off Datasets.

adjacency matrix in all layers, examining if full graph-based attention outperforms PFGA's selective approach and Partially Frozen Attention (PFA): Uses a partially frozen LLMs without graph-based attention, isolating the impact of the adjacency matrix in PFGA. Based on the results as shown in Table IV, PFGA consistently achieves superior accuracy across all metrics and datasets, indicating that the combination of partially frozen layers and selective graph-based attention effectively enhances predictive accuracy. FGA shows higher errors, suggesting that applying graph-based attention at every layer may dilute the model's ability to capture global dependencies. FT performs better than FGA but still falls short of PFGA as it may not optimally leverage pre-trained knowledge, resulting in slightly inferior performance. Even though PFA achieves competitive results, PFGA's selective graph-based attention provides further enhancement by capturing complex spatial dependencies.

G. Parameter Analysis

In the ST-LLM+ framework, depicted in Figure 1, the hyperparameter U is crucial since it determines the number of unfrozen multi-head graph attention layers during the training phase. Figure 3 shows how varying U affects performance across different metrics for the NYCTaxi and CHBike Drop-off datasets. Figure 3 (a) illustrates the performance for the NYCTaxi Drop-off dataset under the WAPE. Initially, the performance improves as U increases to 2, indicating that

TABLE IV: Ablation Study of Partially Frozen Graph-based Attention LLMs.

Datasets	FGA			FT			PFA			PFGA		
	MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE
NYCTaxi Drop-off	5.79	10.53	22.09%	5.68	9.87	24.98%	5.07	9.07	19.18%	**4.94**	**8.68**	**18.86%**
NYCTaxi Pick-up	5.89	10.55	22.26%	5.60	9.83	21.36%	5.29	9.42	20.30%	**5.18**	**8.98**	**19.60%**
CHBike Drop-off	1.94	2.89	39.29%	1.80	2.68	36.33%	1.89	2.81	38.27%	**1.88**	**2.79**	**38.20%**
CHBike Pick-up	2.09	3.28	42.07%	1.91	2.95	38.37%	1.99	3.08	40.19%	**1.98**	**3.05**	**40.11%**

TABLE V: Few-shot Prediction Results on 10% Data of LLM-based Methods.

LLMs	NYCTaxi Pick-up			NYCTaxi Drop-off			CHBike Pick-up			CHBike Drop-off		
	MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE
OFA	6.49	12.12	24.54%	6.27	12.10	23.92%	2.20	3.59	44.40%	2.06	3.17	41.63%
GATGPT	7.02	13.09	26.54%	6.84	13.27	26.09%	2.59	4.41	52.20%	2.50	4.07	50.64%
GCNGPT	10.31	18.82	39.02%	9.25	19.50	35.28%	2.73	4.44	55.20%	2.79	4.65	56.28%
LLaMA-2	5.81	10.16	21.99%	5.59	9.90	21.35%	2.24	3.58	45.20%	2.11	3.23	42.75%
ST-LLM	5.40	9.63	20.45%	5.54	9.84	21.14%	2.07	3.23	41.85%	1.93	2.88	39.21%
ST-LLM+	**5.37**	**9.31**	**19.86%**	**5.46**	**9.77**	**21.12%**	**2.03**	**3.16**	**40.48%**	**1.91**	**2.83**	**39.13%**

TABLE VI: Zero-shot Prediction Results of LLM-based Methods.

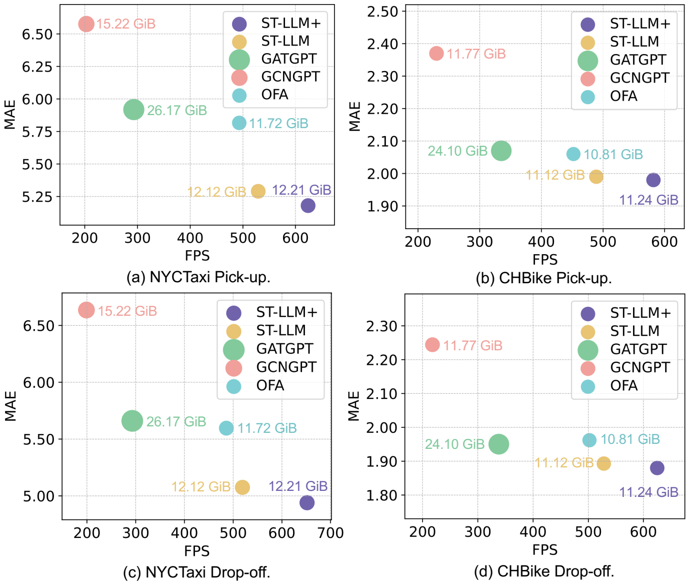
Scenarios	OFA		GATGPT		GCNGPT		LLMAM2		ST-LLM		ST-LLM+	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
NYCTaxi Pick-up → CHBike Drop-off	3.57	5.72	3.25	5.34	3.49	5.64	3.23	5.74	3.12	5.01	**3.03**	**4.82**
NYCTaxi Pick-up → CHBike Pick-up	3.61	5.98	3.29	5.60	3.53	5.91	3.25	5.15	3.06	5.40	**2.87**	**4.73**
NYCTaxi Pick-up → NYCTaxi Drop-off	9.99	20.22	10.00	21.16	11.03	21.86	11.02	22.34	9.31	18.68	**9.03**	**17.93**
NYCTaxi Drop-off → CHBike Drop-off	3.58	5.72	3.19	4.99	3.35	5.19	3.29	4.99	3.09	4.65	**2.79**	**4.52**
NYCTaxi Drop-off → CHBike Pick-up	3.62	5.99	3.26	5.27	3.43	5.49	3.33	5.32	3.02	5.18	**2.94**	**5.01**
NYCTaxi Drop-off → NYCTaxi Pick-up	10.04	17.72	9.67	17.76	8.09	14.58	11.14	20.57	8.02	13.21	**7.93**	**10.85**

TABLE VII: Trainable Parameters (M) Comparisons.

Datasets	Models	All Param.	Trainabl. Param.	Train. %
CHBike	ST-LLM	82.60	42.45	51.40%
	ST-LLM+	82.89	3.12	3.76%
NYCTaxi	ST-LLM	82.60	44.82	54.26%
	ST-LLM+	82.90	5.53	6.67%

unfreezing more layers up to this point enhances the model's accuracy. However, beyond $U = 2$, the performance deteriorates, suggesting diminishing returns or overfitting. In Figure 3 (b), the MAE for the NYCTaxi Drop-off dataset follows a similar trend, where increasing U to 2 leads to reduced errors, highlighting an optimal setting. Beyond this point, however, the MAE rises, reinforcing that $U = 2$ is the optimal balance for minimizing prediction errors. Figure 3 (c) and (d) present the results for the CHBike Drop-off dataset under WAPE and MAE metrics, respectively. In Figure 3 (c), the lowest WAPE is observed at $U = 1$, indicating the peak performance of the model. Similarly, in Figure 3 (d), the MAE is minimized when $U = 1$, after which both metrics worsen as more layers are unfrozen. This suggests that for the CHBike dataset, unfreezing the last layer of LLM is optimal for maintaining model simplicity while achieving the best performance.

H. Efficiency and Trade-off Analysis



(a) NYCTaxi Pick-up.
(b) CHBike Pick-up.
(c) NYCTaxi Drop-off.
(d) CHBike Drop-off.

Fig. 4: Inference Speed and Memory Usage Comparisons.

Figure 4 illustrates the trade-off between inference speed (measured in frames per second, FPS), memory usage (GiB), and MAE for LLM-based methods on the NYCTaxi and CHBike datasets. In this figure, smaller dots represent lower memory usage. For the NYCTaxi dataset, ST-LLM+ achieves the lowest MAE while maintaining a higher FPS and lower memory usage compared to other baselines. Specifically, in the NYCTaxi drop-off dataset, ST-LLM+ increases FPS from 519 to 652 compared to ST-LLM. This improvement highlights the efficiency and effectiveness of the PFGA module in ST-LLM+, demonstrating its advantage over the PFA module in ST-LLM. Although OFA exhibits relatively low memory usage, its FPS remains lower. GATGPT and GCNGPT display lower FPS and higher MAE than ST-LLM+, with GATGPT consuming the most memory and GCNGPT being the slowest. This suggests that integrating GCN with GPT introduces additional computational complexity, whereas GAT's attention mechanism integrates more effectively with LLMs than GCN.

For the CHBike dataset, we observe an overall trend of higher FPS, lower MAE, and reduced memory usage, likely due to the smaller number of stations compared to NYCTaxi. In the CHBike Pick-up dataset, ST-LLM+ once again outperforms all baselines across all metrics, followed closely by ST-LLM, which achieves competitive FPS but slightly lower accuracy. For instance, in the CHBike Pick-up dataset, ST-LLM+ increases FPS from 528 to 654 compared to ST-LLM. OFA maintains its pattern of minimal memory usage while achieving moderate FPS and MAE. GATGPT and GCNGPT continue to exhibit lower FPS and higher MAE. The CHBike Drop-off dataset follows a similar trend, further reinforcing the robustness of ST-LLM+ across different data scenarios. Overall, ST-LLM+ emerges as the most balanced approach, achieving superior inference speed, efficient memory usage, and strong predictive accuracy across both datasets.

Table VII shows the comparison of trainable parameters of the two models, ST-LLM and ST-LLM+. It highlights the significant reduction in trainable parameters achieved by ST-LLM+ compared to its predecessor, ST-LLM, across both CHBike and NYCTaxi datasets. Despite having a slightly higher total number of parameters due to the inclusion of the LoRA-augmented Partially Frozen Graph Attention (PFGA) mechanism, ST-LLM+ dramatically reduces the percentage of trainable parameters, requiring only 3.76% and 6.67% of trainable parameters for CHBike and NYCTaxi, respectively, compared to 51.40% and 54.26% for ST-LLM. This substantial decrease demonstrates the efficiency of the LoRA-based adaptation strategy. This reduces the computational overhead while enhancing the generalization capability of ST-LLM+ by preserving the foundational knowledge from pre-training, making it a scalable and efficient solution

I. Few-Shot Prediction

In few-shot prediction, LLMs are trained with just 10% of the data. The experimental results in Table V demonstrate the strong few-shot learning capabilities of ST-LLM+. From these results, it is evident that ST-LLM+ outperforms other LLM-based models, showcasing its robustness in recognizing complex patterns from limited data. This improvement can be attributed to the effectiveness of the PFGA LLMs, which leverage graph-based attention to capture spatial dependencies even with sparse training data. While ST-LLM demonstrates superior performance in few-shot settings, ST-LLM+ further enhances this capability. For instance, ST-LLM+ achieves a 7.57% reduction in MAE compared to LLaMA-2 on the NYCTaxi Pick-up dataset, while ST-LLM also surpasses LLaMA-2 by 7.06% in MAE. The refinements made in ST-LLM+ contribute to this additional gain, reflecting the impact of the PFGA LLMs and LoRA-augmented fine-tuning.

OFA, GATGPT, and GCNGPT exhibit commendable few-shot performances; however, they fall short of ST-LLM and ST-LLM+ in terms of accuracy. For instance, despite OFA's better performance on the CHBike Drop-off dataset, ST-LLM+ still outperforms it with a 7.28% improvement in MAE. When compared to GATGPT and GCNGPT, ST-LLM+ demonstrates notable average MAE improvements of approximately 22% and 37% across all datasets, respectively, while ST-LLM shows improvements of 21% and 35% over the same models. This difference highlights the advancement ST-LLM+ brings over its predecessor and other LLM-based models, making it a superior choice for few-shot traffic prediction tasks.

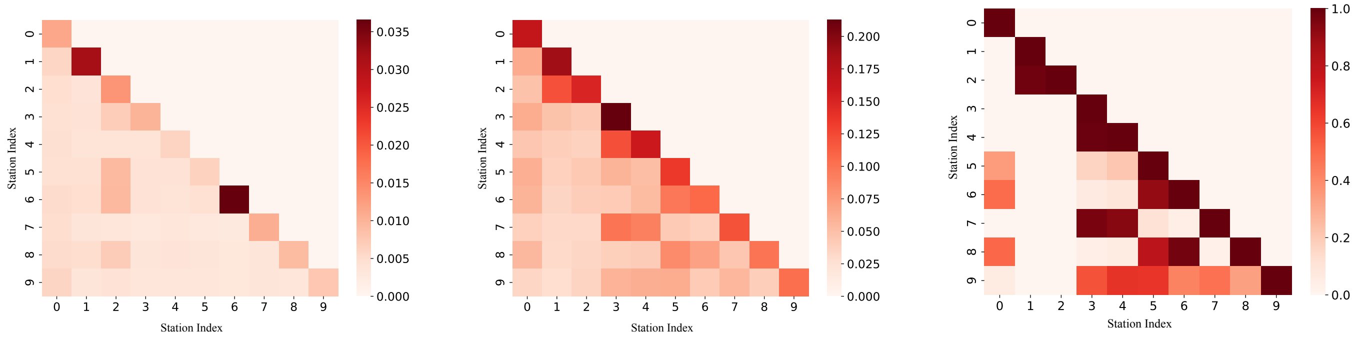
J. Zero-Shot Prediction

The zero-shot prediction experiments evaluate the intra-domain and inter-domain knowledge transfer capabilities of LLMs. Each LLM predicts traffic flow in the CHBike dataset after being trained using only data from the NYCTaxi dataset, without prior exposure to the CHBike data. The results of zero-shot prediction are depicted in Table VI. In terms of intra-domain transfer, such as predicting the NYCTaxi drop-off flow based on the NYCTaxi pick-up flow, ST-LLM+ demonstrates strong accuracy, maintaining low error rates that surpass those of other models, including ST-LLM. This reflects ST-LLM+'s advanced capability to capture and transfer complex spatio-temporal dependencies within the same domain.

ST-LLM+ also excels in inter-domain scenarios, such as transferring from NYCTaxi to CHBike datasets. Compared to other models, ST-LLM+ achieves consistently lower error rates across both MAE and RMSE metrics, demonstrating a robust ability to generalize to new domains without retraining. LLaMA-2 demonstrates strong performance among the baseline models and consistently outperforms other baselines like OFA, GATGPT, and GCNGPT. However, it falls short of matching the superior performance of ST-LLM+ across all evaluated scenarios. The success of ST-LLM+ in zero-shot prediction can be attributed to the PFGA strategy, which enables the model to leverage learned knowledge for intra-domain and inter-domain predictions. With selective graph-based attention, ST-LLM+ adequately captures spatial dependencies and activates the LLM's capabilities for knowledge transfer and reasoning, making it a powerful zero-shot predictor for traffic prediction tasks.

K. Case Study

We conduct a case study to verify the attention patterns in ST-LLM and ST-LLM+ and validate how each model captures spatial dependencies among stations for traffic prediction. Attention maps provide critical insights about the dependencies that each model finds significant, indicating whether it concentrates on broad global dependencies or localized spatial interactions. By comparing the attention maps from ST-LLM



(a) ST-LLM Attention Map.

(b) ST-LLM+ Attention Map.

(c) Adjacency Matrix of NYCTaxi Stations.

Fig. 5: Attention Maps from LLMs and Adjacency Matrix of NYCTaxi Dataset.

and ST-LLM+ against the adjacency matrix of the NYCTaxi dataset, as shown in Figure 5, we aim to illustrate the distinct ways in which each model processes spatial information.

The attention map of ST-LLM in Figure 5 (a) shows a fairly uniform distribution across stations, with attention broadly concentrated along the diagonal. This uniform pattern indicates that ST-LLM primarily captures global dependencies, where each station has a similar level of influence over others without any specific emphasis on spatial proximity. This more generic attention pattern also highlights ST-LLM's lack of ability to prioritize localized spatial dependencies critical for precise traffic prediction. In contrast, ST-LLM+ displays a more specific and structured attention pattern, with distinct attention weights between specific station pairs. For instance, station pairs like (1, 2), (3, 4), and (3, 7) of the NYC Taxi data in Figure 5 (b) show stronger attention, indicating that ST-LLM+ prioritizes certain spatial dependencies based on proximity or correlated traffic patterns. This aligns closely with the corresponding adjacency matrix weights shown in Figure 5 (c), where these station pairs also exhibit strong connections. This alignment between attention weights in ST-LLM+ and adjacency matrix values highlights the model's ability to incorporate spatially relevant connections directly into its attention mechanism, resulting in a more accurate and spatially aware traffic prediction model.

VI. Conclusion

ST-LLM+ achieves progress in calibrating large language models for traffic prediction via spatio-temporal embeddings and partially frozen graph attentions (PFGA). The PFGA enables ST-LLM+ to effectively capture complex spatio-temporal dependencies, enhancing predictive effectiveness and adaptability. The LoRA-augmented training strategy is designed on LLM attention for efficient prediction. Our empirical studies show that ST-LLM+ consistently outperforms state-of-the-art traffic prediction models and other LLM-based approaches, also proving its robustness in few-shot and zero-shot scenarios. In the future, ST-LLM+ can be extended to more tasks, such as traffic data imputation, generation, and anomaly detection, to create a versatile traffic analytics framework.

References

[1] W. Qian, Y. Zhao, D. Zhang, B. Chen, K. Zheng, and X. Zhou, "Towards a unified understanding of uncertainty quantification in traffic flow forecasting," *TKDE*, vol. 36, no. 5, pp. 2239–2256, 2024.

[2] C. Liu, J. Cai, D. Wang, J. Tang, L. Wang, H. Chen, and Z. Xiao, "Understanding the regular travel behavior of private vehicles: An empirical evaluation and a semi-supervised model," *IEEE Sensors Journal*, vol. 21, no. 17, pp. 19 078–19 090, 2021.

[3] G. Jin, C. Liu, Z. Xi, H. Sha, Y. Liu, and J. Huang, "Adaptive dual-view wavenet for urban spatial-temporal event prediction," *Inf. Sci.*, vol. 588, pp. 315–330, 2022.

[4] J. Xia, Y. Yang, S. Wang, H. Yin, J. Cao, and P. S. Yu, "Bayes-enhanced multi-view attention networks for robust POI recommendation," *TKDE*, vol. 36, no. 7, pp. 2895–2909, 2024.

[5] C. Liu, Z. Xiao, W. Long, T. Li, H. Jiang, and K. Li, "Vehicle trajectory data processing, analytics, and applications: A survey," *CSUR*, 2025.

[6] C. Liu, D. Wang, H. Chen, and R. Li, "Study of forecasting urban private car volumes based on multi-source heterogeneous data fusion." *Journal on Communication*, vol. 42, no. 3, 2021.

[7] Y. Fang, Y. Qin, H. Luo, F. Zhao, and K. Zheng, "STWave$^+$: A multi-scale efficient spectral graph attention network with long-term trends for disentangled traffic flow forecasting," *TKDE*, vol. 36, no. 6, pp. 2671–2685, 2024.

[8] H. Miao, J. Shen, J. Cao, J. Xia, and S. Wang, "MBA-STNet: Bayes-enhanced discriminative multi-task learning for flow prediction," *TKDE*, vol. 35, no. 7, pp. 7164–7177, 2023.

[9] S. Yang, Q. Su, Z. Li, Z. Li, H. Mao, C. Liu, and R. Zhao, "Sql-to-schema enhances schema linking in text-to-sql," in *DEXA*, vol. 14910, 2024, pp. 139–145.

[10] X. Du, Z. Li, C. Long, Y. Xing, P. S. Yu, and H. Chen, "FELight: Fairness-aware traffic signal control via sample-efficient reinforcement learning," *TKDE*, vol. 36, no. 9, pp. 4678–4692, 2024.

[11] C. Liu, Z. Xiao, D. Wang, L. Wang, H. Jiang, H. Chen, and J. Yu, "Exploiting spatiotemporal correlations of arrive-stay-leave behaviors for private car flow prediction," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 2, pp. 834–847, 2022.

[12] C. Liu, Z. Xiao, D. Wang, M. Cheng, H. Chen, and J. Cai, "Foreseeing private car transfer between urban regions with multiple graph-based generative adversarial networks," *World Wide Web*, vol. 25, no. 6, pp. 2515–2534, 2022.

[13] C. Zheng, X. Fan, S. Pan, H. Jin, Z. Peng, Z. Wu, C. Wang, and P. S. Yu, "Spatio-temporal joint graph convolutional networks for traffic forecasting," *TKDE*, vol. 36, no. 1, pp. 372–385, 2024.

[14] J. Cai, D. Wang, H. Chen, C. Liu, and Z. Xiao, "Modeling dynamic spatiotemporal user preference for location prediction: a mutually enhanced method," *World Wide Web*, vol. 27, no. 2, p. 14, 2024.

[15] H. Chen, D. Wang, and C. Liu, "Towards semantic travel behavior prediction for private car users," in *HPCC*, 2020, pp. 950–957.

[16] J. Xiao, Z. Xiao, D. Wang, V. Havyarimana, C. Liu, C. Zou, and D. Wu, "Vehicle trajectory interpolation based on ensemble transfer regression," *TITS*, vol. 23, no. 7, pp. 7680–7691, 2022.

[17] Z. Liu, H. Miao, Y. Zhao, C. Liu, K. Zheng, and H. Li, "LightTR: A lightweight framework for federated trajectory recovery," in *ICDE*, 2024.

[18] Z. Shao, Z. Zhang, F. Wang, W. Wei, and Y. Xu, "Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting," in *CIKM*, 2022, pp. 4454–4458.

[19] Y. Jiang, X. Li, Y. Chen, S. Liu, W. Kong, A. F. Lentzakis, and G. Cong, "SAGDFN: A scalable adaptive graph diffusion forecasting network for multivariate time series forecasting," in *ICDE*, 2024.

[20] C. Liu, Z. Xiao, C. Long, D. Wang, T. Li, and H. Jiang, "MVCAR: Multi-view collaborative graph network for private car carbon emission prediction," *TITS*, 2024.

[21] G. Li, S. Zhong, X. Deng, L. Xiang, S.-H. G. Chan, R. Li, Y. Liu, M. Zhang, C.-C. Hung, and W.-C. Peng, "A lightweight and accurate spatial-temporal transformer for traffic forecasting," *TKDE*, vol. 35, no. 11, pp. 10 967–10 980, 2022.

[22] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "iTransformer: Inverted transformers are effective for time series forecasting," in *ICLR*, 2023.

[23] Y. Liu, H. Zhang, C. Li, X. Huang, J. Wang, and M. Long, "Timer: Generative pre-trained transformers are large time series models," in *ICML*, 2024.

[24] D. Ko, J. Choi, H. K. Choi, K. On, B. Roh, and H. J. Kim, "MELTR: meta loss transformer for learning to fine-tune video foundation models," in *CVPR*, 2023, pp. 20 105–20 115.

[25] A. Ramezani and Y. Xu, "Knowledge of cultural moral norms in large language models," in *ACL*, 2023, pp. 428–446.

[26] D. Cao, F. Jia, S. O. Arik, T. Pfister, Y. Zheng, W. Ye, and Y. Liu, "TEMPO: Prompt-based generative pre-trained transformer for time series forecasting," in *ICLR*, 2023.

[27] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan *et al.*, "Time-LLM: Time series forecasting by reprogramming large language models," in *ICLR*, 2024.

[28] C. Liu, S. Yang, Q. Xu, Z. Li, C. Long, Z. Li, and R. Zhao, "Spatial-temporal large language model for traffic prediction," in *MDM*, 2024.

[29] Z. Li, L. Xia, J. Tang, Y. Xu, L. Shi, L. Xia, D. Yin, and C. Huang, "UrbanGPT: Spatio-temporal large language models," in *SIGKDD*, 2024, pp. 5351–5362.

[30] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin, "One Fits All: Power general time series analysis by pretrained lm," in *NeurIPS*, 2023, pp. 1–34.

[31] X. Liu, J. Hu, Y. Li, S. Diao, Y. Liang, B. Hooi, and R. Zimmermann, "UniTime: A language-empowered unified model for cross-domain time series forecasting," in *WWW*, 2024, pp. 4095–4106.

[32] H. Xue and F. D. Salim, "PromptCast: A new prompt-based learning paradigm for time series forecasting," *TKDE*, vol. 36, no. 11, pp. 6851–6864, 2023.

[33] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, "Large language models are zero-shot time series forecasters," *NeurIPS*, vol. 36, 2024.

[34] Y. Liang, H. Wen, Y. Nie, Y. Jiang, M. Jin, D. Song, S. Pan, and Q. Wen, "Foundation models for time series analysis: A tutorial and survey," in *SIGKDD*, 2024, pp. 6555–6565.

[35] H. Wen, Y. Lin, Y. Xia, H. Wan, Q. Wen, R. Zimmermann, and Y. Liang, "DiffSTG: Probabilistic spatio-temporal graph forecasting with denoising diffusion models," in *SIGSPATIAL*, 2023, pp. 60:1–60:12.

[36] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *ICLR*, 2022.

[37] S. Alnegheimish, L. Nguyen, L. Berti-Équille, and K. Veeramachaneni, "Can large language models be anomaly detectors for time series?" in *DSAA*, 2024, pp. 1–10.

[38] Y. Chen, X. Wang, and G. Xu, "GATGPT: A pre-trained large language model with graph attention network for spatiotemporal imputation," *arXiv*, 2023.

[39] C. Chang, W.-C. Peng, and T.-F. Chen, "LLM4TS: Two-stage fine-tuning for time-series forecasting with pre-trained llms," *arXiv*, 2023.

[40] C. Liu, Q. Xu, H. Miao, S. Yang, L. Zhang, C. Long, Z. Li, and R. Zhao, "TimeCMA: Towards llm-empowered time series forecasting via cross-modality alignment," in *AAAI*, 2025.

[41] C. Liu, H. Miao, Q. Xu, S. Zhou, C. Long, Y. Zhao, Z. Li, and R. Zhao, "Efficient multivariate time series forecasting via calibrated language models with privileged knowledge distillation," in *ICDE*, 2025.

[42] J. Li, C. Liu, S. Cheng, R. Arcucci, and S. Hong, "Frozen language model helps ecg zero-shot learning," in *Medical Imaging with Deep Learning*. PMLR, 2024, pp. 402–415.

[43] Z. Pan, Y. Jiang, S. Garg, A. Schneider, Y. Nevmyvaka, and D. Song, "S^{2}IP-LLM: Semantic space informed prompt learning with llm for time series forecasting," in *ICLR*, 2024.

[44] Y. Yuan, J. Ding, J. Feng, D. Jin, and Y. Li, "UniST: a prompt-empowered universal model for urban spatio-temporal prediction," in *SIGKDD*, 2024, pp. 4095–4106.

[45] D. Kieu, T. Kieu, P. Han, B. Yang, C. S. Jensen, and B. Le, "TEAM: Topological evolution-aware framework for traffic forecasting," *PVLDB*, vol. 18, no. 2, pp. 265–278, 2024.

[46] H. Miao, Y. Zhao, C. Guo, B. Yang, Z. Kai, F. Huang, J. Xie, and C. S. Jensen, "A unified replay-based continuous learning framework for spatio-temporal prediction on streaming data," *ICDE*, 2024.

[47] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal arima model with limited input data," *European Transport Research Review*, vol. 7, no. 3, pp. 1–9, 2015.

[48] Z. Lu, C. Zhou, J. Wu, H. Jiang, and S. Cui, "Integrating granger causality and vector auto-regression for traffic prediction of large-scale wlans," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 10, no. 1, pp. 136–151, 2016.

[49] S. Y. Chang, H.-C. Wu, and Y.-C. Kao, "Tensor extended kalman filter and its application to traffic prediction," *TITS*, vol. 24, no. 12, pp. 13 813–13 829, 2023.

[50] B. Shen, X. Liang, Y. Ouyang, M. Liu, W. Zheng, and K. M. Carley, "StepDeep: A novel spatial-temporal mobility event prediction framework based on deep neural network," in *SIGKDD*, 2018, pp. 724–733.

[51] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *AAAI*, vol. 34, no. 01, 2020, pp. 914–921.

[52] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *IJCAI*, 2019, pp. 1907–1913.

[53] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *NeurIPS*, vol. 33, pp. 17 804–17 815, 2020.

[54] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *ICLR*, 2018, pp. 1–16.

[55] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *SIGKDD*, 2019, pp. 1720–1730.

[56] J. Choi, H. Choi, J. Hwang, and N. Park, "Graph neural controlled differential equations for traffic forecasting," in *AAAI*, 2022.

[57] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *IJCAI*, 2018, p. 3634–3640.

[58] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, p. 1907–1913.

[59] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *AAAI*, 2019, pp. 922–929.

[60] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *AAAI*, vol. 34, no. 01, 2020, pp. 1234–1241.

[61] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *TKDE*, vol. 34, no. 11, pp. 5415–5428, 2022.

[62] Z. Lin, M. Li, Z. Zheng, Y. Cheng, and C. Yuan, "Self-attention convlstm for spatiotemporal prediction," in *AAAI*, 2020, pp. 11 531–11 538.

[63] H. Liu, Z. Dong, R. Jiang, J. Deng, J. Deng, Q. Chen, and X. Song, "Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting," in *CIKM*, 2023, pp. 4125–4129.

[64] K. Lu, A. Grover, P. Abbeel, and I. Mordatch, "Frozen pretrained transformers as universal computation engines," in *AAAI*, 2022, pp. 7628–7636.

[65] F. Li, J. Feng, H. Yan, G. Jin, F. Yang, F. Sun, D. Jin, and Y. Li, "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," *TKDD*, vol. 17, no. 1, pp. 9:1–9:21, 2023.

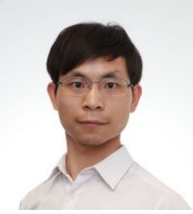
[66] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv*, 2023.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3570705

Chenxi Liu is currently a Research Fellow at the College of Computing and Data Science, Nanyang Technological University (NTU). She got her Ph.D. degree from the College of Computer Science and Electronic Engineering, Hunan University (HNU) in 2023. From 2021 to 2022, she was a visiting PhD at the College of Computing and Data Science, Nanyang Technological University. Her research interests include spatio-temporal data mining, trajectory computing, and large language models.

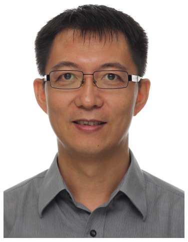
Kethmi Hirushini Hettige received her Bachelor of Science in Industrial Statistics from the University of Colombo, Sri Lanka, in 2020. She is currently pursuing a PhD in Computing and Data Science at Nanyang Technological University, Singapore. Her research interests span spatio-temporal data mining and analytics, interpretable machine learning, large language models (LLMs), and multimodal models with a focus on diverse urban applications, such as air quality monitoring and traffic management.

Qianxiong Xu is currently a Research Fellow at S-Lab, Nanyang Technological University (NTU), working with Prof. Cheng Long. He received his Ph.D degree from NTU, supervised by Prof. Cheng Long, and B.S. degree from Nanjing University of Information Science and Technology in 2020. His research interests include Computer Vision and Spatio-Temporal Data Mining.



Cheng Long is currently an Associate Professor at the College of Computing and Data Science, Nanyang Technological University (NTU). From 2016 to 2018, he worked at Queen's University Belfast, UK. He got his Ph.D. degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology (HKUST) in 2015. His research interests include data management, data mining, and big data analytics.

Shili Xiang received the BE degree from the Department of Computer Science and Technology from University of Science and Technology of China, Hefei, China, in 2003 and the PhD degree in Computer Science from National University of Singapore, Singapore, in 2011. She is currently a principal scientist at the Institute for Infocomm Research (I2R), part of the Agency for Science, Technology and Research, Singapore (A*STAR). Her research interests focus on spatio-temporal data intelligence and its diverse applications, including urban mobility and optimization, AI for transportation, and AI for weather science.



Gao Cong is a professor with the School of Computer Science and Engineering, Nanyang Technological University (NTU). He is the director of Singtel Cognitive and Artificial Intelligence Lab for Enterprises, NTU. Prior to joining NTU, he worked with Aalborg University, Microsoft Research Asia, and the University of Edinburgh. His current research interests include geospatial data management, spatio-temporal data mining, recommendation, and mining social media.

Ziyue Li is an assistant professor with the Information System Department, WiSo Faculty, University of Cologne. He is also the chief machine learning scientist with EWI. His research targets high-dimensional data mining and deep learning methodologies for real-world spatio-temporal problems. His expertises are tensor analysis, spatio-temporal data, and statistical machine learning. Those methods have been applied to various industries, mainly in smart transportation, as well as in smart manufacturing and multimedia. His works have been awarded various Best Paper awards in INFORMS, IISE, and IEEE CASE.

Rui Zhao received the B.S. degree from the University of Science and Technology of China in 2010 and the Ph.D. degree in electronic engineering from The Chinese University of Hong Kong in 2015. He is currently working as a Research Director at SenseTime Research. He is also currently an Adjunct Researcher at the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, the Tsinghua Shenzhen International Graduate School, and the Qing Yuan Research Institute, Shanghai Jiao Tong University. His research interests span a range of topics in computer vision and deep learning, including face recognition, person re-identification, large-scale clustering, unsupervised/self-supervised learning, few-shot/zero-shot learning, and visual-language foundation models.