Spatial-Temporal Large Language Model for Traffic Prediction

Chenxi Liu[1], Sun Yang[2], Qianxiong Xu[1], Zhishuai Li[3], Cheng Long[1,*], Ziyue Li[4,*], Rui Zhao[3]
[1]*S-Lab, Nanyang Technological University, Singapore*
[2]*School of Software and Microelectronics, Peking University, China*
[3]*SenseTime Research, China*
[4]*Information System Department, University of Cologne, Germany*
{chenxi.liu, qianxiong.xu, c.long}@ntu.edu.sg, 2201210484@stu.pku.edu.cn
{lizhishuai, zhaorui}@sensetime.com, zlibn@wiso.uni-koeln.de

***Abstract*—Traffic prediction, an essential component for intelligent transportation systems, endeavours to use historical data to foresee future traffic features at specific locations. Although existing traffic prediction models often emphasize developing complex neural network structures, their accuracy has not improved. Recently, large language models have shown outstanding capabilities in time series analysis. Differing from existing models, LLMs progress mainly through parameter expansion and extensive pretraining while maintaining their fundamental structures. Motivated by these developments, we propose a Spatial-Temporal Large Language Model (ST-LLM) for traffic prediction. In the ST-LLM, we define timesteps at each location as tokens and design a spatial-temporal embedding to learn the spatial location and global temporal patterns of these tokens. Additionally, we integrate these embeddings by a fusion convolution to each token for a unified spatial-temporal representation. Furthermore, we innovate a partially frozen attention strategy to adapt the LLM to capture global spatial-temporal dependencies for traffic prediction. Comprehensive experiments on real traffic datasets offer evidence that ST-LLM is a powerful spatial-temporal learner that outperforms state-of-the-art models. Notably, the ST-LLM also exhibits robust performance in both few-shot and zero-shot prediction scenarios. The code is publicly available at https://github.com/ChenxiLiu-HNU/ST-LLM.**

***Index Terms*—Traffic Prediction, Large Language Model, Spatial-Temporal Data**

I. Introduction

Traffic prediction, which aims to predict future traffic features like traffic flow at specific locations using historical data, is a crucial component for intelligent transportation systems [1]–[7]. This prediction is instrumental in optimizing traffic management [8]–[10] and scheduling public transportation [11]–[13]. For instance, 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 [14]–[16].

The evolution of traffic prediction has seen a shift from traditional time series models to deep learning techniques [17]–[20]. Initially, time series models such as the Autoregressive Integrated Moving Average and Kalman Filter were adapted for their fit with time series data. However, these models are not good at capturing the spatial-temporal dependencies within traffic data, leading to deep learning solutions using convolutional neural networks (CNNs) for spatial and recurrent neural networks (RNNs) for temporal dependencies [18], [21]–[23]. Despite these advancements, the non-Euclidean spatial structure and the complex periodicity of traffic data present challenges for CNNs and RNNs in capturing spatial and temporal dependencies well.

Graph convolutional network (GCN) based models gained popularity for their ability to model local spatial dependencies [3], [24]–[30]. However, these models often encountered over-smoothing issues, which limits their ability to capture global spatial patterns. This shortcoming prompted a shift to attention-based models, which effectively model dynamic spatial correlations without depending on an adjacency matrix [1], [31], [32]. These attention-based approaches have since emerged as a leading trend, offering a superior ability to handle spatial-temporal dependencies in traffic prediction [33], [34]. Nevertheless, with this evolution, the structures of existing traffic prediction models have become progressively complex.

Foundation models, including large language models (LLMs), have advancements in fields such as computer vision [35], [36] and natural language processing [37], [38]. More recently, LLMs also have shown superb performance on time series analysis [39]–[42]. Compared with the complex designs of existing predictive models, LLMs primarily evolve by expanding parameters and pretraining while maintaining their foundational model structure. Existing LLM-based prediction methods focus on the temporal aspect of data in the traffic prediction tasks [41]–[43] and often overlook the spatial aspect. However, in traffic prediction, the spatial variables are strongly correlated and the spatial dimension also proves to be important [44], [45]. For example, a common setting is to use traffic data from the previous *twelve* timesteps to predict traffic for the next *twelve* timesteps at *hundreds* of spatial locations [26] - in this case, more spatial data than temporal data can be leveraged. In our study, we define the timesteps of a spatial location as a token and model the global temporal dependencies across all these tokens to emphasize the spatial aspects.

* Corresponding authors.

Moreover, LLMs are notable for their ability to transfer knowledge across domains [46], such as the pretrained transformer (FPT) LLM [43]. While the FPT LLM is effective in time series analysis tasks, it shows less optimal performance in long-term prediction tasks like traffic prediction. The possible reason is that FPT struggles to bridge the domain gap between language and traffic data. To fill this gap, we propose a partially frozen attention (PFA) LLM specifically designed to enhance traffic prediction accuracy. By partially freezing the multi-head attention layers, the LLM can adapt to traffic prediction while preserving the foundational knowledge acquired during pretraining.

In summary, we propose a novel Spatial-Temporal Large Language Model (ST-LLM) for traffic prediction. Within the ST-LLM framework, we define timesteps at a location as a token. These tokens transform a specialized spatial-temporal embedding layer, which is designed to emphasize spatial locations and global temporal patterns. Furthermore, we fuse the spatial-temporal embeddings of each token for a unified representation. Following this, we introduce the partially frozen attention LLM, a novel strategy tailored for LLMs to capture the global spatial-temporal dependencies in traffic prediction effectively. Extensive experiments on real-world traffic datasets have validated the efficacy of ST-LLM. The key contributions of this paper are summarized as follows:

- We propose a Spatial-Temporal Large Language Model (ST-LLM) for traffic prediction, which defines timesteps at a location as a token and embeds each token by a spatial-temporal embedding layer. We fuse the spatial-temporal embeddings of these tokens uniformly and adapt the LLMs to capture global spatial-temporal dependencies.
- A novel strategy within the LLM, named partially frozen attention, is proposed to enhance the model in traffic prediction. By partially freezing the multi-head attention, the ST-LLM is adapted to capture global spatial-temporal dependencies between tokens for different traffic prediction tasks.
- Extensive experiments are conducted on real traffic datasets to show the superior performance achieved by our ST-LLM across various settings. Moreover, the few-shot and zero-shot prediction results highlight the ST-LLM's capability for intra-domain and inter-domain knowledge transfer.

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

II. Related Work

In this section, we review the related work from two perspectives, large language models for time series analysis and traffic prediction.

A. Large Language Models for Time Series Analysis

Recently large language models (LLMs) have shown superb performance on time series analysis tasks [47], such as prediction [41], classification [42], anomaly detection [43], imputation [48], few-shot learning [49], and zero-shot learning [46]. For instance, TEMPO-GPT combined prompt engineering and seasonal trend decomposition within its generative pretrained transformer (GPT) structure [41]. This integration enabled the model to recall pertinent knowledge from historical data, matching time series inputs with distinct temporal semantic elements. TIME-LLM reprogrammed an LLM for time series forecasting, and the backbone language model remained intact [42]. The authors reprogrammed the input time series with text prototypes before feeding it into the frozen LLM to align the text and time series modalities. OFA employed a frozen GPT2 model across various key tasks in time series analysis [43], the authors concluded that the LLM performed better on tasks of time series, such as imputation, classification, anomaly detection, and few-shot learning. TEST executed time series forecasting and classification tasks [49] and generated the similarity-based, instance-wise, feature-wise, and text-prototype-aligned embedding for time series tokens. LLM-TIME leveraged pretrained LLMs for continuous time series forecasting by representing numbers in text format and generating possible extrapolations through text completions [46]. The above model only models the temporal dimension of the data and ignores the spatial dimension. GATGPT integrates the graph attention network and GPT for spatial-temporal imputation, and the graph attention mechanism boosts the LLM's capability to grasp spatial dependencies [48]. However, it directly overlooks the temporal representation. As of now, the technique for effectively embedding time series data that encompasses both spatial and temporal representations before inputting it into LLMs is not well-defined.

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 [50]–[52]. Traffic data is a special type of time series data. Thus, it is natural to adapt the classic time series models, such as ARIMA and Kalman filter, for the traffic prediction tasks in the early stage [19]. Kumar et al. used the seasonal ARIMA model for short-term traffic prediction [19]. Chang et al. proposed a tensor-extended Kalman filter framework to characterize nonlinear dynamics and applied it to traffic forecasting [2]. However, these models do not perform well due to the inherent spatial-temporal dependencies of traffic data. Later, numerous efforts have been dedicated to advancing traffic prediction techniques by developing various neural network-based models. In the beginning, convolutional neural networks (CNNs) were applied to traffic data to capture spatial dependencies in the data [22]. Shen et al. divided the city into grids and applied 3D CNN for traffic prediction [18]. Yuan et al. used a convolutional long short-term memory network for traffic prediction [18]. 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.

Thanks to the apace of graph learning, graph convolutional network (GCN) based models are popular due to their permutation-invariance, local connectivity, and compositionally [24], [25], [53]. Li et al. modeled the traffic data as a directed graph and introduced a diffusion convolutional recurrent network for traffic prediction [26]. Choi et al. presented a graph neural controlled differential equation for traffic prediction [54]. GCN-based models suffer from oversmoothing, making it hard to capture global spatial dependencies [1]. More recently, attention-based models have emerged as a dominant trend [1], [31]–[34], [55]. Without taking the adjacency matrix into account, attention-based models can still model dynamic spatial correlation more effectively than GCN-based models. In [31], the authors developed an attention-based spatial-temporal graph neural network for traffic prediction. However, the structures of these models are becoming increasingly sophisticated.

III. Problem Definition

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

Definition 2 (Traffic Prediction). Given the historical traffic feature of P timesteps $\mathbf{X}_P = \{\mathbf{X}_{t-P+1}, \mathbf{X}_{t-P+2}, \ldots, \mathbf{X}_t\} \in \mathbb{R}^{P \times N \times C}$, the objective is to learn a function $f(\cdot)$ with parameter θ to predict traffic feature of on the following S timesteps $\mathbf{Y}_S = \{\mathbf{Y}_{t+1}, \mathbf{Y}_{t+2}, \ldots, \mathbf{Y}_{t+S}\} \in \mathbb{R}^{S \times N \times C}$. That is,

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

where each $\mathbf{X}_i \in \mathbb{R}^{N \times C}$.

IV. Methodology

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

A. Overview

The Spatial-Temporal Large Language Model (ST-LLM) framework, as depicted in Figure 1, integrates a spatial-temporal embedding layer, a fusion convolution layer, an LLM layer, and a regression convolution layer. Initially, the historical traffic data is denoted as $\mathbf{X}_P$, which contains N tokens of spatial locations. The $\mathbf{X}_P$ is processed through the spatial-temporal embedding layer, which extracts the token embedding of historical P timesteps, 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 $\mathbf{E}_F$ is input into a PFA LLM that encompasses $L + U$ layers, where the multi-head attention and feed forward layers in the first F layers are frozen to preserve the pretrained knowledge and the multi-head attention layers in the last U layers are unfrozen to enhance the model's focus on capturing the spatial-temporal dependencies between tokens, resulting in the output $\mathbf{H}^L \in \mathbb{R}^{N \times 3D}$. Finally, the regression convolution layer takes $\mathbf{H}^L$ and predicts the following traffic data, denoted as $\widehat{\mathbf{Y}}_S \in \mathbb{R}^{S \times N \times C}$.

B. Spatial-Temporal Embedding and Fusion

We aim to modify LLMs already trained for traffic prediction tasks. We define the timesteps at each location of traffic data as tokens. The spatial-temporal embedding layer transforms the tokens into spatial-temporal representations that align with the LLMs. These representations include spatial correlations, 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 pointwise convolution operation using filters with a 1×1 kernel size. $\mathbf{X}_P$ is the input data, D is the hidden dimension. θ_p represents the learnable parameters of the pointwise convolution.

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

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

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

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

where $\mathbf{W}_{day} \in \mathbb{R}^{T_d \times D}$ and $\mathbf{W}_{week} \in \mathbb{R}^{T_w \times D}$ 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 correlations among token pairs, we design an adaptive embedding of tokens, $\mathbf{E}_S \in \mathbb{R}^{N \times D}$:

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

where σ denotes the activation function, $\mathbf{W}_s \in \mathbb{R}^{D \times D}$ and $\mathbf{b}_s \in \mathbb{R}^D$ are the learnable parameter.

Subsequently, we introduce a fusion convolution (FConv) to project the traffic feature to the required dimensions of the LLM. 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.

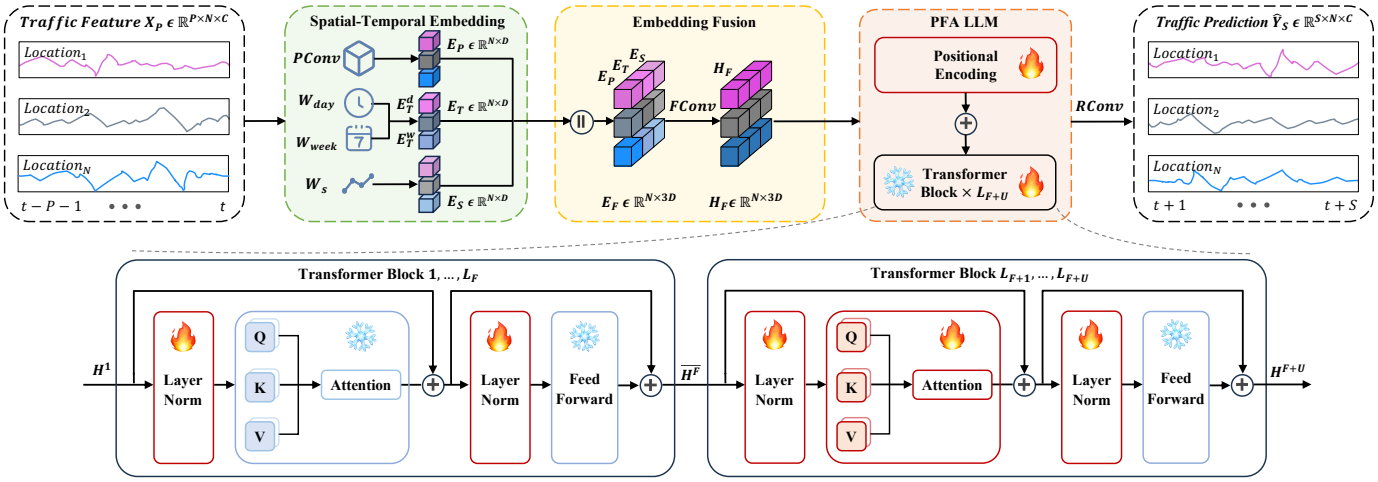


Fig. 1: ST-LLM framework. Given an input traffic feature, we first embed it via a **Spatial-Temporal Embedding**. These embeddings are then integrated uniformly by an **Embedding Fusion** layer. The **PFA (partially frozen attention) LLM** has $F+U$ layers, which are divided into the first F layers and the last U layers. The multi-head attention and feed-forward layers in the first F layers are frozen, and the multi-head attention in the last U layers are unfrozen. The output from PFA LLM is regressed to the prediction results.

C. Partially Frozen Attention (PFA) LLM

The frozen pretrained transformer (FPT) has demonstrated effectiveness in various downstream tasks across non-language modalities [56]. However, its performance is less optimal in tasks requiring short-term and long-term predictions, such as traffic prediction [43]. In this study, we propose a partially frozen attention (PFA) LLM, specifically designed to enhance prediction accuracy in traffic prediction.

The difference between the FPT and our PFA primarily lies in the frozen attention layers. In the FPT framework, both the multi-head attention and feed-forward layers are frozen during training, as these layers contain the most significant portion of the learned knowledge within the LLM. In the PFA, we maintain the first F layers identical to the FPT, but crucially, we unfreeze the last U multi-head attention layers since the attentions effectively handle spatial-temporal dependencies in data. Consequently, our PFA LLM can adapt to traffic prediction while preserving the foundational knowledge acquired during pretraining.

Furthermore, our PFA LLM inverts the traditional calculation dimension from temporal to spatial. This inversion is intentional and aligns with the operation of the partially frozen layers. By focusing on spatial dimensions, our model captures global dependencies more effectively than if we were to concentrate solely on temporal aspects. This shift is particularly relevant in traffic prediction, where spatial dynamics play a critical role in determining flow patterns.

The PFA LLM is built using a Transformer-based architecture, and we choose GPT2 [57]. The GPT2 largely follows the details of the OpenAI GPT model [58] with some modifications. Notably, the layer normalization is positioned at the input of each sub-block, akin to a pre-activation in a residual network. Additionally, an additional layer normalization is added after the final multi-head attention. We visualize these two modifications in the lower part of Figure 1. Furthermore, we introduce a PFA strategy to adapt the GPT2 to capture the spatial-temporal dependencies of the fused tensor $\mathbf{H}_F$.

In the first F layers of the PFA LLM, 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 $\mathbf{H}^1=[\mathbf{H}_F+\mathbf{PE}]$. $\mathbf{PE}$ represent 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) and the first unfrozen layer normalization (LN). $\mathbf{H}^i$ symbolizes the final representation after applying the unfrozen LN and frozen feed-forward network (FFN).

The LN, MHA, and FFN in the PFA LLM are defined as follows:

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

where $\tilde{\mathbf{H}}^i$ is the output of $\mathbf{H}^i$ after passing through the first LN. $\hat{\mathbf{H}}^i$ is the output of $\bar{\mathbf{H}}^i$ after the second LN. 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 to adapt the ST-LLM for capturing spatial-temporal dependencies of traffic data:

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

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. $\mathbf{H}^{F+U}$ denotes the final output of the L_{F+U} layer after applying both the unfrozen LN and frozen FFN, with the MHA being unfrozen.

After the PFA LLM, we design a regression convolution (RConv) to predict the traffic features on the following S timesteps:

$$\hat{\mathbf{Y}}_S = \mathit{RConv}(\mathbf{H}^{F+U}; \theta_r), \tag{11}$$

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:

$$\mathcal{L} = \left\|\widehat{\mathbf{Y}}_S - \mathbf{Y}_S\right\| + \lambda \cdot L\text{reg}, \tag{12}$$

where $\widehat{\mathbf{Y}}_S$ is the predicted traffic feature. $\mathbf{Y}_S$ is the ground truth. Lreg represents the L2 regularization term, which helps control overfitting. λ is a hyperparameter. The whole process of the ST-LLM is shown in Algorithm 1.

Algorithm 1: The ST-LLM Framework

Input: Traffic feature $\mathbf{X}_P$ in the historical P timesteps, and all hyperparameters.
Output: Trained ST-LLM.

for *each epoch* **do**
  Shuffle training data
  for *each batch* $\mathbf{X}_P$ *in training data* **do**
    $\mathbf{E}_F \leftarrow$ Spatial-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 LLM 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 (11).
    Update all learnable parameters by minimizing the loss in Equation (12) with $\widehat{\mathbf{Y}}_S$ and $\mathbf{Y}_S$ via Ranger21 optimizer.
  end
end

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], CHBike[2].

TABLE I: Dataset Description.

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

NYCTaxi. The 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 timesteps, 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. The dataset aligns with NYCTaxi in terms of time, covering 4,368 timesteps with each timestep representing a 30-minute interval.

B. Baselines

We compare ST-LLM with the following 10 baselines belonging to three categories: (1) GNN-based models: DCRNN [26], STGCN [27], GWN [25], AGCRN [24], STG-NCDE [54], DGCRN [3]. (2) Attention-based models: AST-GCN [34], GMAN [33], ASTGNN [31]. (3) LLMs: OFA [43], GATGPT [48], GCNGPT, and LLAMA2. The details of the baselines are outlined as follows:

- DCRNN [26]: it models the data as a directed graph, and introduces diffusion convolutional recurrent network.
- STGCN [27]: A graph convolutional network that combines 1D convolution to tackle the time series prediction task in the traffic domain.
- GWN [25]: A graph neural network that employs graph convolution with an adaptive adjacency matrix.
- AGCRN [24]: it is an adaptive graph convolutional recurrent network that incorporates node learning and inter-dependency inference among traffic series.
- STG-NCDE [54]: it presents the graph neural controlled differential equation for processing sequential data.
- DGCRN [3]: it introduces a traffic prediction framework using dynamic graph convolutional recurrent networks.

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

- ASTGCN [34]: it is an attention-based spatial-temporal graph convolutional network for traffic forecasting.
- GMAN [33]: it is an attention-based predictive model that adopts an encoder-decoder architecture.
- ASTGNN [31]: it is an attention-based model for learning the dynamics and heterogeneity of traffic data.
- OFA [43]: it refrains from altering the self-attention and feed-forward networks of the residual blocks in the GPT2. We take an inverted view on traffic data of OFA for better prediction performance.
- GATGPT [48]: it combines the GAT with the frozen pretrained transformer GPT2.
- GCNGPT: it combines the GCN with the frozen pretrained transformer GPT2.
- LLAMA2: it is a collection of pretrained and fine-tuned large language models developed by Meta. In the LLAMA2, we adapt the frozen pretrained transformer.

C. Implementations

Aligning with contemporary practices, we divided the NYC-Taxi and CHBike datasets into training, validation, and test sets using a 6:2:2 ratio. We set the historical timesteps P and the future timesteps S to 12 each, enabling multi-step traffic prediction. 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 GPT2 and LLAMA2 7B. We configured GPT2 with six layers [43], and LLAMA2 with eight layers [42]. The batch size is 64 and the max training epoch is 500. Note that the experimental results are averaged across all prediction timesteps.

D. Evaluation Metrics

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

$$\text{MAE} = \frac{1}{m}\sum_{i=1}^{m}\left|\widehat{\mathbf{Y}}_i - \mathbf{Y}_i\right|, \quad \text{MAPE} = \frac{100\%}{m}\sum_{i=1}^{m}\left|\frac{\widehat{\mathbf{Y}}_i - \mathbf{Y}_i}{\mathbf{Y}_i}\right|, \tag{13}$$

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\widehat{\mathbf{Y}}_i - \mathbf{Y}_i\right)^2}, \ \text{WAPE} = \frac{\sum_{i=1}^{m}\left|\widehat{\mathbf{Y}}_i - \mathbf{Y}_i\right|}{\sum_{i=1}^{m}|\mathbf{Y}_i|}\times 100\%, \tag{14}$$

where m is the number of all predicted values.

E. Main Results

The comparison results with baselines are shown in Table II. The bold results are the best, and the underlined results are the second best. The LLM in ST-LLM is GPT2. We can make the following observations. (1) LLM-based methods yield superior prediction results, with ST-LLM exhibiting the most effective performance. The ST-LLM outperforms other LLMs in four traffic prediction scenarios, demonstrating its superior accuracy in handling diverse traffic data across various datasets. (2) OFA and LLAMA2 are competent but surpassed by ST-LLM which achieves a 22.5% average MAE improvement over OFA and 20.8% over LLAMA2. This may be due to OFA's ineffective traffic data embedding, making it difficult for LLMs to understand the spatial-temporal dependencies between data. Despite LLAMA2's larger size and complexity, it doesn't directly translate to better traffic prediction than ST-LLM. GATGPT and GCNGPT do not extract temporal representations of traffic data to influence the LLM to capture spatial-temporal dependencies. (3) Attention-based models like ASTGNN and GMAN exhibit varied performance across different datasets. They performed quite well in some cases but were always inferior to ST-LLM. This variability could be attributed to the limitations of traditional attention mechanisms in handling complex spatial-temporal embeddings, especially when compared to large language models. (4) GNN-based Models such as GWN and DGCRN demonstrate competitive performance, particularly in specific metrics, but still cannot outperform ST-LLM. This suggests that while GNNs effectively capture spatial dependencies, their temporal analysis capabilities might not be as advanced as the ST-LLM, which limits their overall performance.

In summary, the experimental results from the traffic prediction tasks using the NYCTaxi and CHBike datasets demonstrate a clear trend in performance among different types of models. LLMs-based methods emerge as the top performers, showcasing their superior ability to handle different traffic prediction tasks. Following LLM-based models, attention-based models occupy the second tier. Finally, GCN-based models, while still effective, rank lower compared to the aforementioned models. This hierarchy highlights the evolving landscape of model capabilities, with LLM-based methods leading the way in traffic prediction tasks.

F. Performance of ST-LLM and Ablation Studies

With or Without Different Components. The ST-LLM comprises several crucial components, each contributing to its overall effectiveness in traffic prediction. This section compares variants of ST-LLM concerning the following aspects to investigate the effectiveness of different components. w/o LLM: A variant of ST-LLM with the LLM being removed. w/o ST: A variant of ST-LLM with the spatial-temporal embedding being removed. w/o T: A variant of ST-LLM with the temporal embedding being removed. w/o S: A variant of ST-LLM with the spatial embedding being removed.

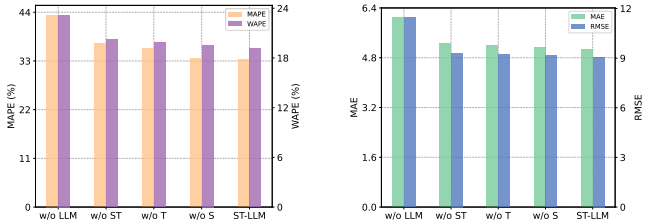
Figure 2 presents the ablation study on the NYCTaxi Pick-up and Drop-off datasets, examining the impact of different components in the ST-LLM. The w/o LLM variant shows a considerable increase in error across all metrics. Its removal leads to a degradation in performance, demonstrating that the prediction capabilities of the ST-LLM are heavily reliant on the

TABLE II: Model comparison on NYC datasets in terms of MAE, RMSE, MAPE (%), and WAPE (%). Results are averaged from all prediction timesteps.

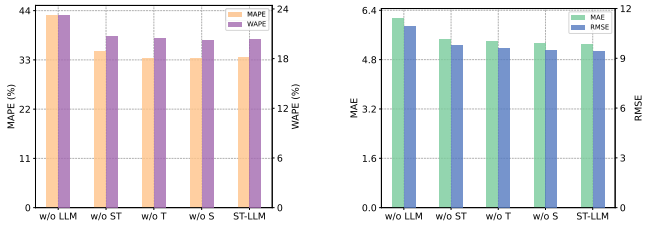
Dataset	NYCTaxi Pick-up				NYCTaxi Drop-off				CHBike Pick-up				CHBike Drop-off			
Metric	MAE	RMSE	MAPE	WAPE	MAE	RMSE	MAPE	WAPE	MAE	RMSE	MAPE	WAPE	MAE	RMSE	MAPE	WAPE
DCRNN	5.40	9.71	35.09%	20.43%	5.19	9.63	37.78%	19.82%	2.09	3.30	54.22%	42.26%	1.96	2.94	51.42%	39.61%
STGCN	5.71	10.22	36.51%	21.62%	5.38	9.60	39.12%	20.55%	2.08	3.31	53.63%	42.08%	2.01	3.07	50.45%	40.62%
ASTGCN	7.43	13.84	47.96%	28.04%	6.98	14.70	45.48%	26.60%	2.76	4.45	64.23%	55.71%	2.79	4.20	69.88%	56.49%
GWN	5.43	**9.39**	37.79%	20.55%	**5.03**	**8.78**	35.63%	19.21%	2.04	3.20	**53.08%**	40.95%	1.95	2.98	50.30%	39.43%
AGCRN	5.79	10.11	40.40%	21.93%	5.45	9.56	40.67%	20.81%	2.16	3.46	56.35%	43.69%	2.06	3.19	51.91%	41.78%
GMAN	5.43	9.47	34.39%	20.42%	5.09	8.95	35.00%	19.33%	2.20	3.35	57.34%	44.06%	2.09	3.00	54.82%	42.00%
STSGCN	6.19	11.14	39.67%	25.37%	5.62	10.21	37.92%	22.59%	2.36	3.73	58.17%	50.09%	2.73	4.50	57.89%	54.10%
ASTGNN	5.90	10.71	40.15%	22.32%	6.28	12.00	49.78%	23.97%	2.37	3.67	60.08%	47.81%	2.24	3.35	57.21%	45.27%
STG-NCDE	6.24	11.25	43.20%	23.46%	5.38	9.74	40.45%	21.37%	2.15	3.97	55.49%	61.38%	2.28	3.42	60.96%	46.06%
DGCRN	5.44	9.82	35.78%	20.58%	5.14	9.39	35.09%	19.64%	2.06	3.21	54.06%	41.51%	1.96	2.93	51.99%	39.70%
OFA	5.82	10.42	36.67%	22.00%	5.60	10.14	37.39%	21.36%	2.06	3.21	53.55%	41.70%	1.96	2.97	49.64%	39.68%
GATGPT	5.92	10.55	37.83%	22.39%	5.66	10.39	37.36%	21.60%	2.07	3.23	52.54%	41.70%	1.95	2.94	49.26%	39.43%
GCNGPT	6.58	12.23	40.19%	24.88%	6.64	12.24	42.46%	25.32%	2.37	3.80	56.24%	47.66%	2.24	3.48	51.05%	45.37%
LLAMA2	5.35	9.48	41.32%	20.27%	5.66	10.74	47.47%	21.63%	2.10	3.37	56.63%	42.49%	1.99	3.03	55.23%	40.28%
ST-LLM	**5.29**	9.42	**33.55%**	**20.03%**	5.07	9.07	**33.34%**	**19.18%**	**1.99**	**3.08**	53.54%	**40.19%**	**1.89**	**2.81**	**49.50%**	**38.27%**

TABLE III: Ablation Study of Partially Frozen Attention LLM.

LLM	No Pretrain			Full Layer			Full Tuning			FPT			PFA		
Metric	MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE
NYCTaxi Drop-off	5.22	9.19	19.93%	5.22	9.33	19.91%	5.90	10.36	22.51%	5.73	10.43	21.87%	**5.07**	**9.07**	**19.18%**
NYCTaxi Pick-up	5.36	9.39	20.27%	5.43	9.65	20.54%	5.98	10.40	22.63%	5.83	10.45	22.04%	**5.29**	**9.42**	**20.30%**
CHBike Drop-off	1.92	2.86	38.84%	1.91	2.83	38.63%	1.90	2.82	38.28%	1.92	2.86	38.90%	**1.89**	**2.81**	**38.27%**
CHBike Pick-up	2.03	3.14	40.87%	2.02	3.12	40.62%	2.01	3.11	40.43%	2.07	3.25	41.65%	**1.99**	**3.08**	**40.19%**

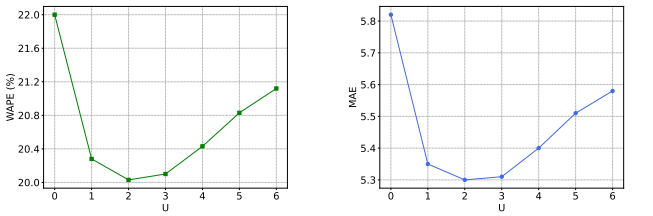


(a) Drop-off under MAPE and WAPE. (b) Drop-off under MAE and RMSE.

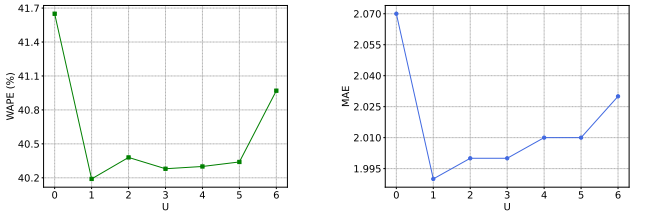


(c) Pick-up under MAPE and WAPE. (d) Pick-up under MAE and RMSE.

Fig. 2: Ablation study of ST-LLM on NYCTaxi dataset.



(a) NYCTaxi Pick-up under WAPE. (b) NYCTaxi Pick-up under MAE.



(c) CHBike Pick-up under WAPE. (d) CHBike Pick-up under MAE.

Fig. 3: Performance study of unfreezing last U layers.

LLM's ability to learn complex dependencies from traffic data. The exclusion of spatial-temporal embedding (w/o ST) results in a notable performance drop. This highlights the importance of spatial-temporal embedding in understanding the spatial-temporal dependencies within the context of traffic data. The experimental results reveal that removing either temporal (w/o T) or spatial (w/o S) components similarly affects the model's prediction error. Ablating either of these embeddings results in an increased error, indicating the necessity of both for accurate predictions. Notably, the model incurs a larger prediction error without the temporal component, underscoring the significance of our thoughtfully designed hour-of-day and day-of-week embeddings. This observation further emphasizes the critical role that balanced spatial and temporal embeddings play in enhancing the model's predictive performance. We observe the lowest error rates across all metrics when all components are integrated, as in the full ST-LLM model. This underscores the effect of combining LLM, spatial, and temporal embeddings to handle the spatial-temporal dependencies of traffic prediction.

Ablation Study of Partially Frozen Attention LLM. In this subsection, we conducted an ablation study to evaluate the efficacy of our proposed Partially Frozen Attention (PFA) LLM. The PFA is compared against several variations: Frozen Pretrained Transformer (FPT), models without pretraining (No Pretrain), models utilizing the full twelve layers of GPT-2

(Full Layer), and fully tuned models without any frozen layers (Full Tuning). The ablation results of PFA LLM are shown in table III. The PFA demonstrates superior performance across all metrics on all datasets. This suggests that partially freezing the attention significantly enhances the predictive accuracy. The FPT shows commendable performance, it is slightly outperformed by the PFA. This indicates that the partial freezing strategy strikes a more optimal balance between leveraging prelearned features and adapting to new data. The Full Layer and Full Tuning models exhibit competitive performance. However, they still fall short of the efficiency and accuracy demonstrated by the PFA model. This underscores the advantage of selective freezing in managing the adaptability of the model. The comparison with the No Pretrain model highlights the significant role of pretraining in model performance. While the No Pretrain model performs reasonably well, it is evident that pretraining, especially when combined with strategies like partial frozen, is crucial for achieving higher levels of accuracy.

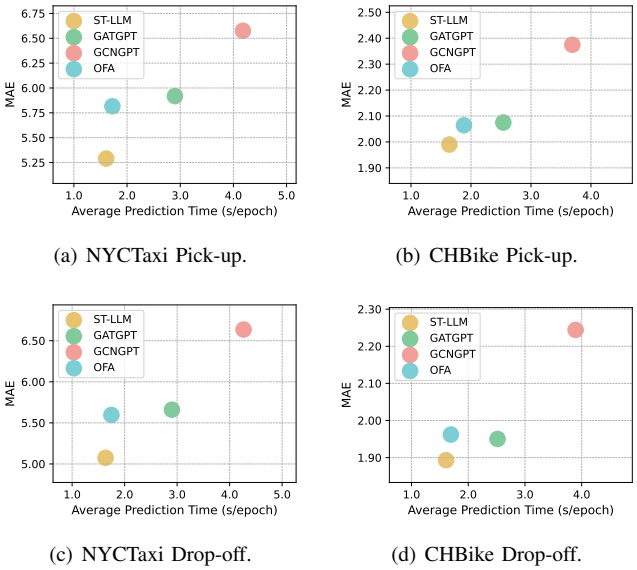
G. Parameter Analysis

In the ST-LLM framework shown in Figure 3, the hyperparameter U plays a pivotal role in determining the count of unfrozen multi-head attention layers throughout the training phase. As depicted in Figure 3 (a), for the NYCTaxi Pick-up dataset, the performance, as measured by WAPE, initially improves with the increase of U to 2. The trend suggests that unfreezing additional layers up to a certain threshold can enhance the performance of the ST-LLM. However, this positive effect inverts when U exceeds 2, at which point the model's performance starts to degrade, hinting at the diminishing benefits associated with unfreezing more layers. Figure 3 (b) presents a consistent pattern, where the MAE for the NYCTaxi Pick-up dataset decreases with an increase in U to 2. In a nutshell, the optimal U of taxi flow prediction is set to 2.

For the CHBike Pick-up dataset, as shown in Figure 3 (c), setting U to 1 results in the lowest WAPE, signifying the peak performance of the model. An increase in U leads to a rise in WAPE, which signals a decline in accuracy. Figure 3 (d) illustrates a similar pattern on the CHBike Pick-up dataset. The MAE indicates an increase in performance as U is set to 1, with the lowest MAE observed at this value. This reinforces the observation that a single unfrozen multi-head attention layer is optimal for minimizing absolute errors, and the model achieves the balance between complexity and performance. This optimal point suggests that unfreezing more layers does not contribute to improved accuracy and might even degrade ST-LLM performance.

H. Inference Time Analysis

Figure 4 illustrates the trade-off between inference time and MAE for various LLMs on NYCTaxi and CHBike datasets. It's important to note from the outset that LLAMA2 is intentionally omitted from this comparative analysis because its inference time is significantly longer than that of the other



(a) NYCTaxi Pick-up. (b) CHBike Pick-up.

(c) NYCTaxi Drop-off. (d) CHBike Drop-off.

Fig. 4: Inference time of LLMs.

LLMs under consideration, making it an outlier in terms of efficiency. For NYCTaxi datasets, ST-LLM achieves the lowest MAE while maintaining competitive inference times. Following closely is OFA, whose inference time is similar to ST-LLM, albeit with a slightly higher MAE. This suggests that spatial-temporal embedding and PFA do not slow down the inference speed of the LLM, yet enhance prediction accuracy. GATGPT and GCNGPT exhibit longer inference times and higher MAEs than ST-LLM, with GCNGPT being the slowest. This could be attributed to the fact that while GCN generally has a simpler structure, their combination with GPT might introduce extra computational complexity. This also implies that GAT's attention mechanism could be more efficient when combined with GPT.

For CHBike datasets, we observe that the inference times and MAEs are generally lower than those for the NYCTaxi datasets, which might indicate differences in the datasets' complexity of the prediction tasks. For the CHBike Pick-up dataset, ST-LLM again achieves the lowest MAE. OFA follows with a closely competitive inference time but again falls slightly short on accuracy. GATGPT and GCNGPT show a consistent pattern, having longer inference times and higher MAEs, with GCNGPT being the slowest model. The CHBike Drop-off dataset reflects a similar pattern, which shows ST-LLM's robustness across different data scenarios. The OFA model remains a close second in speed, with a negligible increase in MAE. The trend of longer inference times for GATGPT and GCNGPT persists, with GCNGPT again taking the longest time among the models. In summary, ST-LLM stands out as the model providing the best balance between inference speed and predictive accuracy across both datasets.

TABLE IV: Few-shot prediction results on 10% data of LLMs.

LLM	NYCTaxi Pick-up				NYCTaxi Drop-off				CHBike Pick-up				CHBike Drop-off			
	MAE	RMSE	MAPE	WAPE	MAE	RMSE	MAPE	WAPE	MAE	RMSE	MAPE	WAPE	MAE	RMSE	MAPE	WAPE
OFA	6.49	12.12	46.74%	24.54%	6.27	12.10	45.23%	23.92%	2.20	3.59	57.52%	44.40%	2.06	3.17	55.96%	41.63%
GATGPT	7.02	13.09	50.19%	26.54%	6.84	13.27	56.15%	26.09%	2.59	4.41	56.23%	52.20%	2.50	4.07	56.36%	50.64%
GCNGPT	10.31	18.82	59.41%	39.02%	9.25	19.50	56.77%	35.28%	2.73	4.44	56.93%	55.20%	2.79	4.65	61.85%	56.28%
LLAMA2	5.81	10.16	41.82%	21.99%	5.59	9.90	40.58%	21.35%	2.24	3.58	59.47%	45.20%	2.11	3.23	54.44%	42.75%
ST-LLM	**5.40**	**9.63**	**33.36%**	**20.45%**	**5.54**	**9.84**	**39.56%**	**21.14%**	**2.07**	**3.23**	**55.68%**	**41.85%**	**1.93**	**2.88**	**52.75%**	**39.21%**

TABLE V: Zero-shot prediction results of LLMs.

LLM	OFA			GATGPT			GCNGPT			LLMAM2			ST-LLM		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
NYCTaxi Pick-up → CHBike Drop-off	3.57	5.72	59.26%	3.25	5.34	59.35%	3.49	5.64	59.06%	3.23	5.74	72.14%	**3.12**	**5.01**	**55.12%**
NYCTaxi Pick-up → CHBike Pick-up	3.61	5.98	59.55%	3.29	5.60	59.71%	3.53	5.91	59.14%	3.25	5.15	88.52%	**3.06**	**5.40**	**50.94%**
NYCTaxi Pick-up → NYCTaxi Drop-off	9.99	20.22	75.14%	10.00	21.16	68.03%	11.03	21.86	70.32%	11.02	22.34	94.31%	**9.31**	**18.68**	**66.42%**
NYCTaxi Drop-off → CHBike Drop-off	3.58	5.72	59.33%	3.19	4.99	76.75%	3.35	5.19	69.36%	3.29	4.99	80.87%	**3.09**	**4.65**	**52.73%**
NYCTaxi Drop-off → CHBike Pick-up	3.62	5.99	59.55%	3.26	5.27	79.41%	3.43	5.49	71.76%	3.33	5.32	82.60%	**3.02**	**5.18**	**68.27%**
NYCTaxi Drop-off → NYCTaxi Pick-up	10.04	17.72	88.10%	9.67	17.76	73.46%	8.09	14.58	50.99%	11.14	20.57	94.03%	**8.02**	**13.21**	**46.16%**

I. Few-Shot Prediction

In few-shot prediction, LLMs are trained with just 10% of data. The experimental results are in Table IV. From the results, we can see that ST-LLM is superior in recognizing complex patterns from limited data, and we attribute this to the knowledge activation in our PFA LLM. While the LLAMA2 model presents competitive results, especially on the NYCTaxi datasets. However, it does not consistently surpass the performance of ST-LLM. For instance, on the NYCTaxi Pick-up dataset, ST-LLM achieves a noteworthy 7.06% reduction in MAE compared to LLAMA2. The OFA, GATGPT, and GCNGPT, although commendable in their performances, do not match the superior results of ST-LLM. Notably, despite OFA's better performance on the CHBike Drop-off dataset, ST-LLM still outperforms it with a 9.15% improvement in MAE. Compared with GATGPT and GCNGPT, ST-LLM shows remarkable average improvements of over 39.21% and 7.80% in MAE across all datasets, respectively. This significant difference highlights the robustness of ST-LLM in efficiently handling scenarios with limited data.

J. Zero-Shot Prediction

The zero-shot prediction experiments evaluate the intra-domain and inter-domain knowledge transfer capabilities of various LLMs. Each LLM in this evaluation predicts traffic flow in the CHBike dataset after being trained using only data from the NYCTaxi dataset, without prior exposure to the CHBike dataset. The results of zero-shot prediction are depicted in Table V. In terms of intra-domain transfer, such as predicting the NYCTaxi drop-off flow based on the NYCTaxi pick-up flow, ST-LLM demonstrates its ability to maintain high accuracy. The results also show that ST-LLM exhibits exceptional performance in inter-domain scenarios, such as transferring from NYCTaxi datasets to CHBike datasets. The ST-LLM consistently achieves the lowest error rates, indicating a robust ability to adapt to new domains without retraining. The results proved that the OFA is not a good zero-shot predictor. GATGPT and GCNGPT show competent adaptability but still fall short compared to ST-LLM's performance, particularly in challenging inter-domain transfers. LLAMA2 performs well in most inter-domain scenarios, and its performance is second only to ST-LLM. In conclusion, the zero-shot prediction results reinforce the adaptability and predictive strength of ST-LLM. We attribute this success to our PFA strategy being better at activating the LLM's knowledge transfer and reasoning capabilities when performing traffic prediction tasks.

VI. Conclusion

ST-LLM shows promise in adapting large language models for traffic prediction by embedding traffic data into spatial-temporal representations for LLMs. A partially frozen attention strategy is proposed to adapt the LLM to capture global spatial-temporal dependencies in traffic prediction. Our empirical studies show that the proposed ST-LLM performs better than the state-of-the-art traffic prediction models and LLMs. Future work will explore LLM for multi-task learning, such as incorporating traffic imputation, generation, and anomaly detection.

Acknowledgment

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

References

[1] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *AAAI*, 2023, pp. 4365–4373.

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

[3] 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," *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 1, pp. 9:1–9:21, 2023.

[4] J. Gong, Y. Liu, T. Li, H. Chai, X. Wang, J. Feng, C. Deng, D. Jin, and Y. Li, "Empowering spatial knowledge graph for mobile traffic prediction," in *SIGSPATIAL*, 2023, pp. 1–11.

[5] 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.

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

[7] Q. Xu, S. Ruan, C. Long, L. Yu, and C. Zhang, "Traffic speed imputation with spatio-temporal attentions and cycle-perceptual training," in *CIKM*, 2022, pp. 2280–2289.

[8] 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.

[9] Z. Zhou, J. Shi, H. Zhang, Q. Chen, X. Wang, H. Chen, and Y. Wang, "Crest: A credible spatiotemporal learning framework for uncertainty-aware traffic forecasting," in *WSDM*, 2024, pp. 1–10.

[10] 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.

[11] G. Jin, Y. Liang, Y. Fang, Z. Shao, J. Huang, J. Zhang, and Y. Zheng, "Spatio-temporal graph neural networks for predictive learning in urban computing: A survey," *IEEE Trans. Knowl. Data Eng.*, pp. 1–20, 2023.

[12] S. Wang, H. Miao, H. Chen, and Z. Huang, "Multi-task adversarial spatial-temporal networks for crowd flow prediction," in *CIKM*, 2020, p. 1555–1564.

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

[14] A. Liu and Y. Zhang, "Spatial-temporal dynamic graph convolutional network with interactive learning for traffic forecasting," *IEEE Trans. Intell. Transp. Syst.*, 2024.

[15] 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.

[16] Z. He, J. Zhang, C. Chow, N. Li, X. Liu, P. Lin, and X. Sun, "Pairwise and hyper-correlations based spatiotemporal neural networks for traffic speed predictions," in *MDM*, 2023, pp. 235–244.

[17] D. Campos, M. Zhang, B. Yang, T. Kieu, C. Guo, and C. S. Jensen, "LightTS: Lightweight time series classification with adaptive ensemble distillation," *SIGMOD*, vol. 1, no. 2, pp. 171:1–171:27, 2023.

[18] Z. Yuan, X. Zhou, and T. Yang, "Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data," in *SIGKDD*, 2018, pp. 984–992.

[19] 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.

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

[21] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis, and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4927–4943, 2021.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

[28] H. Miao, X. Zhong, J. Liu, Y. Zhao, X. Zhao, W. Qian, K. Zheng, and C. S. Jensen, "Task assignment with efficient federated preference learning in spatial crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, 2023.

[29] X. Zhong, H. Miao, D. Qiu, Y. Zhao, and K. Zheng, "Personalized location-preference learning for federated task assignment in spatial crowdsourcing," in *CIKM*, 2023, pp. 3534–3543.

[30] Q. Xu, C. Long, Z. Li, S. Ruan, R. Zhao, and Z. Li, "Kits: Inductive spatio-temporal kriging with increment training strategy," *arXiv preprint arXiv:2311.02565*, 2023.

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] Z. Yu, S. Wu, Y. Fu, S. Zhang, and Y. C. Lin, "Hint-aug: Drawing hints from foundation vision transformers towards boosted few-shot parameter-efficient tuning," in *CVPR*, 2023, pp. 11 102–11 112.

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

[38] J. Maynez, P. Agrawal, and S. Gehrmann, "Benchmarking large language model capabilities for conditional generation," in *ACL*, 2023, pp. 9194–9213.

[39] H. Xue, B. P. Voutharoja, and F. D. Salim, "Leveraging language foundation models for human mobility forecasting," in *SIGSPATIAL*, 2022, pp. 1–9.

[40] H. Xue and F. D. Salim, "Promptcast: A new prompt-based learning paradigm for time series forecasting," *IEEE Trans. Knowl. Data Eng.*, pp. 1–14, 2023.

[41] 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.

[42] 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.

[43] 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.

[44] M. Lablack and Y. Shen, "Spatio-temporal graph mixformer for traffic forecasting," *Expert Systems with Applications*, vol. 228, p. 120281, 2023.

[45] 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.

[46] S. Q. Nate Gruver, Marc Finzi and A. G. Wilson, "Large language models are zero shot time series forecasters," in *NeurIPS*, 2023, pp. 1–29.

[47] M. Jin, Q. Wen, Y. Liang, C. Zhang, S. Xue, X. Wang, J. Zhang, Y. Wang, H. Chen, X. Li *et al.*, "Large models for time series and spatio-temporal data: A survey and outlook," *arXiv*, 2023.

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

[49] C. Sun, Y. Li, H. Li, and S. Hong, "Test: Text prototype aligned embedding to activate llm's ability for time series," in *ICLR*, 2023.

[50] J. Ye, L. Sun, B. Du, Y. Fu, and H. Xiong, "Coupled layer-wise graph convolution for transportation demand prediction," in *AAAI*, 2021, pp. 4617–4625.

[51] Z. Shao, Z. Zhang, W. Wei, F. Wang, Y. Xu, X. Cao, and C. S. Jensen, "Decoupled dynamic spatial-temporal graph neural network for traffic forecasting," *VLDB*, vol. 15, no. 11, p. 2733–2746, 2022.

[52] H. Miao, J. Shen, J. Cao, J. Xia, and S. Wang, "Mba-stnet: Bayes-enhanced discriminative multi-task learning for flow prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 7164–7177, 2023.

[53] 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.

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

[55] H. Xue, F. D. Salim, Y. Ren, and C. L. Clarke, "Translating human mobility forecasting through natural language generation," in *WSDM*, 2022, pp. 1224–1233.

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

[57] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[58] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," *OpenAI blog*, 2018.