LLM4ST-Traffic: Leveraging Large Language Models for Cross-modal Knowledge Transfer to Overcome Data Sparsity in Traffic Prediction

Anonymous ACL submission

Abstract

Traffic prediction is a core challenge of Intelligent Transportation Systems (ITS). The development of deep learning has driven significant advancements in traffic prediction models; however, the increased complexity of these models has led to higher demands for data scale. Existing models have encountered performance bottlenecks due to an imbalance between excessive complexity and data sparsity. This paper proposes LLM4ST-Traffic, a traffic prediction framework based on Large Language Models (LLMs), aimed at addressing data scarcity through cross-modal semantic alignment and lightweight fine-tuning. The core innovations include: the Cross-Modal Alignment (CMA) module, which utilizes cross-attention to establish deep connections between traffic features (such as flow trends and periodicity) and textual concepts, thereby overcoming the semantic disjunction caused by traditional linear projections, and Prefix Adapter Fine-Tuning (PAFT), which implements learnable prefix prompts for lightweight training, optimizing predictive performance while retaining pre-trained knowledge. Experimental results indicate that LLM4ST-Traffic demonstrates exceptional performance in prediction accuracy and robustness, exhibiting outstanding performance in low-sample scenarios. Interpretability analysis validates the effectiveness of semantic alignment.

1 Introduction

Traffic prediction, as a core component of intelligent urban perception systems, aims to infer future road network states from historical traffic data (such as traffic volume and speed). Its accuracy directly impacts the effectiveness of downstream tasks such as traffic signal control and route planning. Despite significant advancements in traffic prediction models (Yu et al., 2017; Wu et al., 2019) due to developments in deep learning, the resulting model complexity has intensified the dependence on large-scale data. However, the traffic domain often faces the challenge of data sparsity: hardware costs limit sensor deployment, leading to insufficient spatial coverage, and sudden events cause temporal distribution shifts. Existing models, which rely on predefined road network biases and massive training data, are prone to overfitting in low-resource scenarios, exposing the fundamental contradiction between data scale and model performance.

The recent open-domain generalization capabilities of large language models (LLMs) provide new insights for addressing this issue (Lu et al., 2022), but their transferability in traffic prediction faces critical bottlenecks. Existing methods, such as TPLLM (Ren et al., 2024) and STLLM (Liu et al., 2024a), employ static linear projections to map traffic data into text space, achieving only numerical and formal alignment without establishing semantic-level associations. Moreover, the fine-tuning strategies exhibit severe imbalances: STLLM risks knowledge forgetting by adjusting 50% of the parameters, while conservative fine-tuning limits performance improvements.

To tackle these challenges, this paper proposes the LLM4ST-Traffic framework, with its core innovation being the construction of a semantics-driven cross-modal collaborative paradigm. First, we design the Cross-Modal Alignment (CMA) module, which utilizes a cross-attention mechanism to achieve interactive mapping between traffic features and text concepts, replacing traditional static projections to address the modality gap. Second, we introduce the Prefix Adapter Fine-Tuning (PAFT) strategy, which designs a hierarchical, learnable prefix adaptation module that fine-tunes only 1.31% of the LLM parameters, optimizing prediction performance while retaining pre-trained knowledge. Experimental results show that this framework reduces the Mean Absolute Error (MAE) by an average of 11.7% compared to

TPLLM, STLLM, and GATGPT across four real-world datasets, demonstrating a 5.5% performance advantage in low-sample scenarios (10% training data). Visualization of the attention module further validates the effectiveness of semantic alignment. The main contributions of this paper include:

- This paper proposes LLM4ST-Traffic, the first LLM-based 'decoupling-semantic alignment-adaptation' technical system for traffic prediction, which provides a new solution for data-sparse scenarios.
- The design of the CMA module, which overcomes the limitations of static mapping, and the introduction of the PAFT strategy, achieves a balance between performance and retention of pre-trained knowledge.
- Experiments conducted on real traffic datasets demonstrate the outstanding performance of LLM4ST-Traffic, and it also shows excellent performance in few-shot learning scenarios. Additionally, an interpretability analysis was performed.

2 Related Work

In this section, we will discuss traffic prediction from three aspects: Traffic Prediction, LLMs for Traffic Prediction, and LLMs for Spatio-Temporal Traffic Prediction.

2.1 Traffic Prediction

The core challenge in traffic prediction lies in the strong coupling of spatio-temporal features: the road network topology forms rigid spatial constraints, and the dynamic changes of traffic flow increase the difficulty of temporal modeling. Early statistical models (e.g., ARIMA) (Hamed et al., 1995) and traditional machine learning methods (e.g., SVM, KNN) (Ding et al., 2002; Zheng and Su, 2014) rely on handcrafted features and linear assumptions, making it difficult to capture complex nonlinear relationships. Deep learning approaches break through traditional limitations by decoupling spatio-temporal modules. They utilize Graph Convolutional Networks (GCN) (Kipf and Welling, 2016) to model the spatial structure of the road network and combine Recurrent Neural Networks. (e.g., GRU, LSTM) (Graves and Graves, 2012; Cho et al., 2014) or Temporal Convolutional Networks (TCN) (Bai et al., 2018) to capture the dynamic evolution of traffic flow. Typical examples are two-stream architectures such as STGCN (Yu et al., 2017) and T-GCN (Zhao et al., 2019). Attention mechanism models (e.g., ASTGCN, GMAN) (Guo et al., 2019; Zheng et al., 2020) further enhance the ability to model spatiotemporal dependencies through dynamic weight allocation. However, with the increase in model complexity, their demand for data scale also increases significantly. The current contradiction is that complex models are prone to overfitting in low-resource scenarios with sparse sensors and frequent unexpected events (Emmert-Streib et al., 2020), highlighting the fundamental bottleneck between the limitations of data scale and the improvement of model performance in traditional paradigms.

2.2 LLMs For Time Series Analysis

In recent years, large language models (LLMs) have transferred open-domain knowledge to the field of time series analysis through parameter-efficient fine-tuning (PEFT) (Han et al., 2024), giving rise to two types of solutions for data sparsity. Feature extraction methods such as GPT4TS (Zhou et al., 2023) directly map time series into the text embedding space, while TIME-LLM (Jin et al., 2023) innovatively converts numerical segments into pseudo-text tokens such as 'rising trend' and uses the attention mechanism of frozen LLMs to improve few-shot reasoning capabilities; CALF (Liu et al., 2024b) takes another approach by enhancing cross-modal representations through text-time series contrastive learning. In the direction of Prompt engineering (Zhang et al., 2024), TEMPO-GPT (Cao et al., 2023) encodes time series patterns into natural language templates, and PromptCast (Xue and Salim, 2023) automatically generates prompts suitable for complex scenarios through dynamic instructions.

2.3 LLMs For Traffic Perdition

In the field of spatiotemporal traffic prediction, when attempting to integrate large language models (LLMs) (Devlin, 2018; Radford et al., 2019; Brown et al., 2020; Touvron et al., 2023), challenges of cross-modal alignment and imbalance in knowledge transfer arise. GATGPT (Chen et al., 2023) integrates graph attention networks to extract road network topologies, yet it neglects the dynamic evolution over time. TPLLM (Ren et al., 2024) alleviates data sparsity through spatio-temporal dual embeddings and LoRA (Hu et al., 2021) fine-tuning but is constrained by the static representations of linear mappings. STLLM (Liu et al., 2024a) adopts

node sequence tokenization and frozen attention fine-tuning. Although it improves long-term prediction capabilities, the strong fine-tuning strategy leads to changes in 50 % of the pre-trained model's parameters, posing the risk of semantic knowledge forgetting. The common limitations of existing methods lie in that cross-modal interactions rely on one-way concatenation or static mapping, lack semantic-level fusion; and fine-tuning strategies struggle to strike a balance between over-adjustment (damaging generalization) and under-adjustment (limiting performance). Based on this, this paper constructs a novel LLM-based traffic prediction framework.

3 Problem Definition

This section describes the characteristics of traffic data and defines the problem.

Traffic Features: We represent the traffic feature data as a tensor $\mathbf{X} \in \mathbb{R}^{T \times N \times C}$, where T denotes the number of time steps, N is the number of nodes, and C represents the feature dimensions.

Traffic Prediction: Given historical traffic data $\mathbf{X}_P = \{\mathbf{X}_{t-P+1}, \mathbf{X}_{t-P+2}, \ldots, \mathbf{X}_t\} \in \mathbf{X} \in \mathbb{R}^{P \times N \times C}$ on P time steps, the goal is to learn a function $f(\cdot)$ with parameters θ to predict the subsequent S time steps $\mathbf{Y}_S = \{\mathbf{Y}_{t+1}, \mathbf{Y}_{t+2}, \ldots, \mathbf{Y}_{t+S}\} \in \mathbb{R}^{S \times N \times C}$. Formally, this can be expressed as:

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

where each $X_i \in \mathbb{R}^{N \times S}$.

4 Methodology

In this section, the details and components of LLM4ST-Traffic are described.

4.1 Overview

As illustrated in Figure 1, the LLM4ST-Traffic framework performs traffic prediction in data-sparse scenarios through four distinct stages. First, the multi-granularity spatio-temporal embedding layer extracts temporal patterns, periodic trends, and spatial topology features, integrating them into a unified representation. Next, the Cross-Modal Alignment (CMA) module employs a cross-attention mechanism to map traffic feature data into the semantic space of a pre-trained language model, thereby achieving alignment between different modalities. Subsequently, the framework utilizes a Prefix Adapter Fine-Tuning (PAFT) strategy to fine-tune the pre-trained LLM, adapting it to the specific requirements of traffic prediction tasks. Finally, the regression layer projects the semantic features into the prediction space to generate multi-step future traffic states.

4.2 Spatio-Temporal Embedding

To ensure semantic compatibility between traffic data and pre-trained language models, we have designed a multi-granularity spatial-temporal embedding architecture comprising three core components: Patch Embedding, Time Embedding, and Node Embedding. To accommodate the input shape required by LLMs, we collapse the time step dimension into the feature dimension within the embedding module.

Patch Embedding To resolve the conflict between the discreteness of single time-step data and the semantic continuity required by language models, we propose a sliding window-based temporal semantic aggregation method. Given an input sequence $\mathbf{X} \in \mathbb{R}^{T \times N \times C}$ (where T is the number of time steps, N is the number of nodes, and C is the feature dimension), we construct temporal patches for each node as follows:

$$\mathbf{P}_i = \mathrm{Concat}(\mathbf{X}_{[t:t+k],i}) \in \mathbb{R}^{k \times C} \tag{2}$$

Here, k denotes the patch size (default $k = 3$), $i \in \{1, ..., N\}$. These patches are then mapped to the semantic space aligned with LLM word embeddings through a linear projection:

$$\mathbf{E}_P = f_{\text{patch}}(\mathbf{P}) = \mathbf{P}\mathbf{W}_p + \mathbf{b}_p \in \mathbb{R}^{N \times D} \tag{3}$$

In this equation, $\mathbf{W}_p \in \mathbb{R}^{(k \cdot C) \times D}$ represents the learnable parameters, and D is one-third of the LLM's word embedding dimensionality.

Time Embedding To explicitly model the periodic characteristics of traffic flow, we design a dual-scale time encoding that captures both daily and weekly patterns. Specifically, daily patterns are encoded using a learnable matrix $\mathbf{E}_{\text{daily}} \in \mathbb{R}^{24 \times D}$ to represent hourly variations, while weekly patterns are captured through $\mathbf{E}_{\text{weekly}} \in \mathbb{R}^{7 \times D}$ to represent weekly cycles. For an input timestamp t, the time embedding is computed as:

$$\mathbf{E}_T = \mathbf{E}_{\text{daily}}[h(t)] + \mathbf{E}_{\text{weekly}}[d(t)] \in \mathbb{R}^D \tag{4}$$

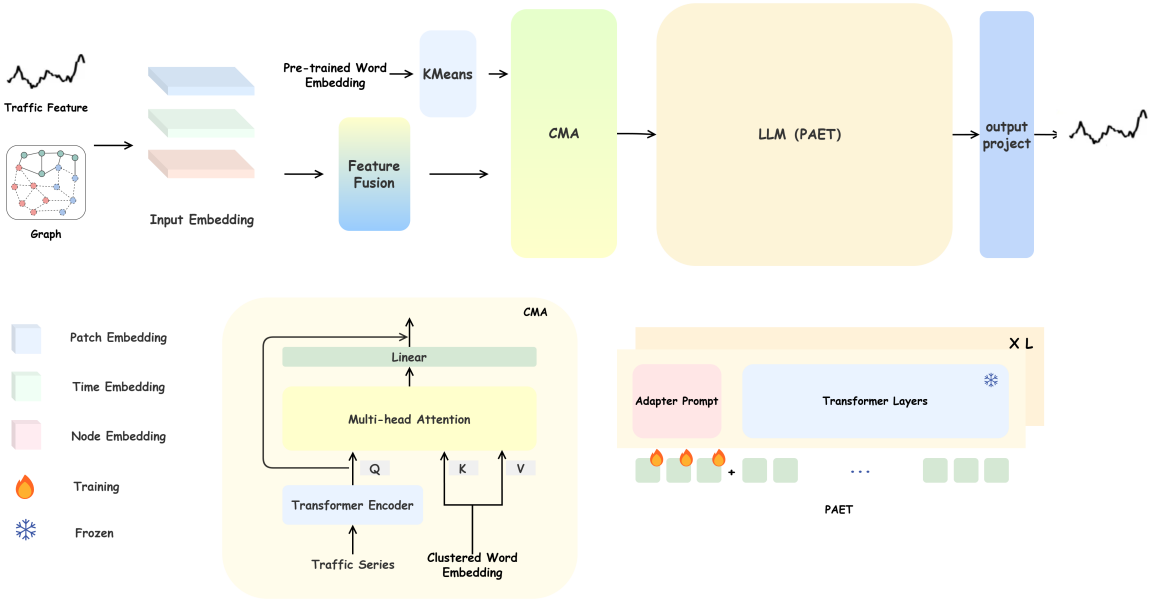

Figure 1: The model framework of LLM4ST-Traffic. The upper section features an overall architecture diagram, while the lower section provides detailed specifics. CMA refers to the Cross-Modality Alignment module, and PAFT refers to the Prefix Adapter Fine-Tuning module.

where $h(t) \in \{0, ..., 23\}$ denotes the hour index and $d(t) \in \{0, ..., 6\}$ denotes the day of the week index. After broadcasting to all nodes, the resulting time embedding is $\mathbf{E}_T \in \mathbb{R}^{N \times D}$.

Node Embedding To capture the spatial dependencies of the road network, we design an adaptive node embedding matrix:

$$\mathbf{E}_S = \mathbf{E}_{\text{node}} \in \mathbb{R}^{N \times D} \tag{5}$$

where $\mathbf{E}_{\text{node}}$ is a learnable parameter initialized uniformly.

Fusion Feature The fusion feature concatenates the three sets of embeddings along the feature dimension to generate a joint representation compatible with LLMs:

$$\mathbf{E}_{\text{final}} = \text{Concat}(\mathbf{E}_P, \mathbf{E}_T, \mathbf{E}_S) \in \mathbb{R}^{N \times 3D} \tag{6}$$

4.3 Cross-Model Alignment

To achieve dynamic alignment between traffic spatio-temporal features and the semantic space of pre-trained language models (LLMs), we propose the Cross-Modal Alignment (CMA) module.

Feature Enhancement The input spatiotemporal embeddings $\mathbf{E}_{\text{final}} \in \mathbb{R}^{B \times N \times 3D}$ (where B is the batch size, N is the number of nodes, and $3D$ the feature dimension) are processed through a TransformerEncoder layer to enhance the context-aware capabilities of the spatiotemporal features:

$$\mathbf{E}_{\text{ctx}} = \text{TransformerEncoder}(\mathbf{E}_{\text{final}}) \tag{7}$$

Here,$\mathbf{E}_{\text{norm}} \in \mathbb{R}^{B \times N \times 3D}$, $\mathbf{E}_{\text{ctx}} \in \mathbb{R}^{B \times N \times D_{\text{llm}}}$, $D_{\text{llm}} = 768$. denotes the dimension of the LLM word vectors. The encoder consists of 2 Transformer layers with a default of 8 attention heads.

Semantic Clustering Considering that the semantic information of traffic data is relatively simple and that the vocabulary of large language models (LLMs) typically contains tens of thousands of tokens, directly aligning them would lead to a waste of computational resources. Therefore, we employ the K-Means clustering method to perform dimensionality reduction on the pre-trained word embedding matrix $\mathbf{W}_{\text{vocab}} \in \mathbb{R}^{|\mathcal{V}| \times D_{\text{llm}}}$:

$$\hat{\mathbf{W}}_{\text{vocab}} = \text{KMeans}(\mathbf{W}_{\text{vocab}}, d) \in \mathbb{R}^{|\mathcal{V}| \times d} \tag{8}$$

where $d = 500$. By identifying semantically similar word groups to form "synonym clusters," we significantly reduce the computational complexity from $\mathcal{O}(N \cdot |\mathcal{V}|$ to $\mathcal{O}(N \cdot d)$ while preserving key semantics through the aggregation of synonym clusters.

Dynamic Attention Alignment Using a multi-head cross-attention mechanism, we establish a soft alignment between spatiotemporal features and

semantic terms:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D_{\text{llm}}}}\right)\mathbf{V} \tag{9}$$

Here, $\mathbf{Q}$ is derived from the enhanced spatiotemporal features, and both $\mathbf{K}, \mathbf{V}$ are obtained from the vocabulary embeddings clustered by the Kmeans method.

Residual Feature Fusion To fully leverage the information from the original spatiotemporal features, we introduce residual connections and a Multi-Layer Perceptron (MLP) to enhance the model's non-linearity:

$$\mathbf{Z}_{\text{out}} = \text{MLP}(\mathbf{Z}) + \mathbf{E}_{\text{ctx}} \in \mathbb{R}^{N \times D_{\text{llm}}} \tag{10}$$

This design ensures that important spatiotemporal information is not lost during the alignment process.

4.4 Prefix Adapter Fine-Tuning

To address the challenging trade-off between improving model performance and mitigating knowledge forgetting, we propose the Prefix-based Efficient Tuning method. The core idea is to prepend learnable prefix prompts to the input data, guiding the pre-trained model to adapt effectively to the prediction task. Trainable prefixes are concatenated to the input of each layer of the LLM, generated through the following steps: First, perform embedding initialization with trainable prompt vectors for each layer $\mathbf{P}^{(l)} \in \mathbb{R}^{m \times D_{\text{llm}}}$, where the default $m = 30$ is the prefix length and $D_{\text{llm}} = 768$. Then, add trainable positional encoding $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{m \times D_{\text{llm}}}$to enhance sequence position awareness, formulated as: $\mathbf{H}^{(l)}_{\text{prompt}} = \mathbf{P}^{(l)} + \mathbf{E}_{\text{pos}}$. Finally, apply a lightweight MLP for non-linear projection to further enhance the expressiveness of the prefixes, resulting in $\tilde{\mathbf{P}}^{(l)} = \text{MLP}(\mathbf{H}^{(l)}_{\text{prompt}})$.

The generated prefix prompts are concatenated with the input data as $\tilde{\mathbf{H}}^{(l)} = \text{Concat}(\tilde{\mathbf{P}}^{(l)}, \mathbf{H}^{(l)})$, and then fed into the TransformerLayer of the LLM. During the output phase, the original input segment is extracted as $(\mathbf{H}^{(l)} = \mathbf{H}^{(l)}[:, m:, :])$, ensuring that subsequent layers are not affected by the prefix. From an implementation perspective, we employ an adaptation prompt module to independently generate prefixes for each layer, thereby enabling lightweight fine-tuning of the LLM. Considering that the aligned token embeddings differ from those in the original LLM, we intentionally retain the trainability of the normalization layers.

4.5 Regression Layer

Use a linear layer to map high-dimensional semantic features into the prediction space to forecast traffic conditions for the next T time steps:

$$\hat{\mathbf{Y}} = \mathbf{H}_{\text{out}}\mathbf{W}_r + \mathbf{b}_r \in \mathbb{R}^{N \times T} \tag{11}$$

where $\mathbf{H}_{\text{out}} \in \mathbb{R}^{N \times D_{\text{llm}}}$ represents the output of the LLM, $\mathbf{W}_r \in \mathbb{R}^{D_{\text{llm}} \times T}$ is the learnable weight matrix and $\mathbf{b}_r \in \mathbb{R}^T$ is the bias term.

5 Experiments

In this section, we aim to validate the superiority of our LLM4ST-Traffic model through a series of comprehensive experimental evaluations.

5.1 Experimental Setup

Datasets Our approach is extensively evaluated on four real-world spatio-temporal benchmark datasets: METR-LA, PEMS-BAY, PEMS04, and PEMS08. The first two datasets, METR-LA and PEMS-BAY, were introduced in the DCRNN (Li et al., 2017), while PEMS04 and PEMS08 were proposed in the STSGCN (Song et al., 2020). All four datasets have a temporal resolution of five minutes, resulting in 12 timesteps per hour. Table 1 provides further details of these datasets.

Dataset	Sensors	Timesteps	Time Range
METR-LA	207	34,272	03/2012 - 06/2012
PEMS-BAY	325	52,116	01/2017 - 05/2017
PEMS04	307	16,992	01/2018 - 02/2018
PEMS08	170	17,856	07/2016 - 08/2016

Table 1: Summary of Datasets.

Implementation: For dataset splitting, we employed different ratios: the METR - LA and PEMS - BAY datasets were divided into training, validation, and test sets in a 7:1:2 ratio, respectively, while the PEMS04 and PEMS08 datasets were split using a 6:2:2 ratio. In addition, regarding model configuration, both the input sequence length and prediction sequence length were set to one hour (T = T' = 12 timesteps). The model was trained using the Adam optimizer with an initial learning rate of 0.001 and a learning rate decay strategy. During training, the batch size was set to 64. For the LLM component, we selected GPT-2 as the base model, utilizing its first three transformer layers. All experiments were conducted and evaluated on a Linux server equipped with an NVIDIA RTX 4090 GPU.

Datasets		Metric	HI	DCRNN	AGCRN	STGCN	MTGNN	STNorm	GMAN	PDFormer	GATGPT	STLLM	**ours**
METR-LA	Horizon3 (15min)	MAE	6.80	2.67	2.85	2.75	2.69	2.81	2.80	2.83	2.89	2.92	**2.64**
		RMSE	14.21	5.16	5.53	5.29	5.16	5.57	5.55	5.45	5.49	5.55	**5.09**
		MAPE	16.72%	6.86%	7.63%	7.10%	6.89%	7.40%	7.41%	7.77%	7.45%	7.53%	**6.76%**
	Horizon6 (30 min)	MAE	6.80	3.12	3.20	3.15	3.05	3.18	3.12	3.20	3.28	3.24	**2.99**
		RMSE	14.21	6.27	6.52	6.35	6.13	6.59	6.49	6.46	6.53	6.49	**6.10**
		MAPE	16.72%	8.42%	9.00%	8.62%	8.16%	8.47%	8.73%	9.19%	8.94%	8.86%	**8.13%**
	Horizon12 (60 min)	MAE	6.80	3.54	3.59	3.60	3.47	3.57	3.44	3.62	3.73	3.61	**3.37**
		RMSE	14.20	7.47	7.45	7.43	7.21	7.51	7.35	7.49	7.65	7.45	**7.17**
		MAPE	10.15%	10.32%	10.47%	10.35%	**9.70%**	10.24%	10.07%	10.91%	10.62%	10.37%	9.82%
PEMS-BYA	Horizon3 (15min)	MAE	3.06	1.31	1.35	1.36	1.33	1.33	1.35	1.32	1.35	1.35	**1.29**
		RMSE	7.05	**2.76**	2.88	2.88	2.80	2.82	2.90	2.83	2.82	2.84	**2.76**
		MAPE	6.85%	2.73%	2.91%	2.86%	2.81%	2.76%	2.87%	2.78%	2.85%	2.79%	**2.68%**
	Horizon6 (30 min)	MAE	3.06	1.65	1.67	1.70	1.66	1.65	1.65	1.64	1.69	1.66	**1.60**
		RMSE	7.04	3.75	3.82	3.84	3.77	3.77	3.82	3.79	3.82	3.76	**3.71**
		MAPE	6.84%	3.71%	3.81%	3.79%	3.75%	3.66%	3.74%	3.71%	3.79%	3.67%	**3.54%**
	Horizon12 (60 min)	MAE	3.05	1.97	1.94	2.02	1.95	1.92	1.92	1.91	2.00	1.96	**1.87**
		RMSE	7.03	4.60	4.50	4.63	4.50	4.45	4.49	4.43	4.58	4.47	**4.35**
		MAPE	6.83%	4.68%	4.55%	4.72%	4.62%	4.46%	4.52%	4.51%	4.62%	4.50%	**4.31%**

Table 2: Performance on METR-LA and PEMS-BAY.

Dataset	PEMS04			PEMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HI	42.35	61.66	29.92 %	36.66	50.45	21.63 %
DCRNN	19.63	31.26	13.59 %	15.22	24.17	10.21 %
AGCRN	19.38	31.25	13.40 %	15.32	24.41	10.03%
STGCN	19.57	31.38	13.44 %	16.08	25.39	10.60%
MTGCN	19.17	31.70	13.37 %	15.18	24.24	10.20 %
STNorm	18.96	30.98	12.69 %	15.41	24.77	9.76%
GMAN	19.14	31.60	13.19 %	15.31	24.92	10.13 %
ASTGCN	21.83	34.48	14.25 %	18.33	28.30	11.64%
GATGPT	22.77	34.65	19.22 %	18.33	27.38	17.72%
TPLLM	19.53	31.91	12.81 %	15.45	25.35	9.88 %
STLLM	21.41	32.39	18.41 %	17.98	26.82	15.26 %
ours	**18.49**	**30.01**	**12.20** %	**14.09**	**23.65**	**9.15** %

Table 3: Performance on PEMS04 and PEMS08.

Baselines In this study, we compare our proposed method with several widely used baseline models in the field. Among these, HI is a classic traditional model (Cui et al., 2021). We also consider DCRNN (Li et al., 2017), AGCRN (Bai et al., 2020), STGCN (Yu et al., 2017), and MTGNN (Wu et al., 2020), all of which leverage graph-related information for modeling. Additionally, we examine STNorm (Deng et al., 2021), which focuses on the decomposition of traffic time series. For mainstream attention-based architectures, we include ASTGCN (Guo et al., 2019), GMAN (Zheng et al., 2020), and PDFormer (Jiang et al., 2023). In the context of integrating Large Language Models (LLMs) into traffic prediction, we evaluate GATGPT (Chen et al., 2023), STLLM (Liu et al., 2024a), and TPLLM (Ren et al., 2024).

Evaluation Metrics We employ three commonly used metrics to assess the performance of the proposed framework: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). For all metrics, lower values indicate superior predictive performance. The computation processes for the evaluation metrics are as follows:

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{m}\sum_{i=1}^{m}\left|\widehat{\mathbf{Y}}_i - \mathbf{Y}_i\right| \\ \mathrm{MAPE} &= \frac{100\%}{m}\sum_{i=1}^{m}\left|\frac{\widehat{\mathbf{Y}}_i - \mathbf{Y}_i}{\mathbf{Y}_i}\right| \\ \mathrm{RMSE} &= \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\widehat{\mathbf{Y}}_i - \mathbf{Y}_i\right)^2} \end{aligned} \tag{12}$$

where m is the number of all predicted values.

5.2 Overall Performance

We investigated the predictive capabilities of the LLM4ST-Traffic model.Table 2 and Table 3 present the comparative results with baseline models on the METR-LA and PEMS-BAY datasets, as well as the PEMS04 and PEMS08 datasets, respectively. Bolded results indicate the best performance. It is important to note that the TP-LLM code was not publicly available; therefore, we directly utilized the results provided in its original paper. In Table III, we present the mean of the predictions over 12 time steps as the final displayed results. The findings clearly show that LLM4ST-Traffic exhibits superior performance across all datasets. A detailed analysis is provided in the appendix A.

5.3 Ablation Study

Component Ablation Figure 2 presents an ablation study on the METR-LA and PEMS-BAY datasets, aiming to evaluate the impact of different components within the LLM4ST-Traffic model. LLM4ST-Traffic comprises several key components, each playing a crucial role in the overall effectiveness of traffic forecasting. This section examines the effectiveness of each component by comparing the following variants:

- **w/o CMA**: Variant without the Cross-Modal Alignment Module.
- **w/o LLM**: Variant without the pretrained Large Language Model.
- **w/o Patch**: Variant without patch embedding, using a simple linear mapping instead.

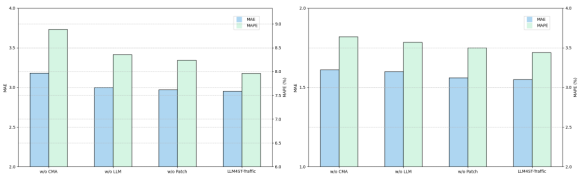

Figure 2: Ablation experiments on METR-LA and PEMS-BAY.

Main Observations: Removing the pre-trained model (w/o LLM) and using only the multi-granular embedding layer and alignment module results in a significant decline in model performance. This indicates that the LLM plays a key role in enhancing predictive performance by leveraging its strengths in semantic understanding and feature extraction. Similarly, removing the CMA module (w/o CMA) leads to a notable decrease in performance, demonstrating that this module is essential for aligning the semantics between traffic data and the pre-trained language model, effectively handling the alignment between different data modalities. When the patch embedding component is removed (w/o Patch), the model's prediction metrics increase, suggesting that the model's predictive capability relies on the temporal semantic aggregation method within the patch embedding. Overall, when all components (patch embedding, alignment module, and LLM) are integrated, the model achieves the lowest error rates across all metrics. This further validates the effectiveness of these components in handling traffic forecasting tasks and demonstrates the superior performance of LLM4ST-Traffic through the synergistic interaction of its components.

LLM	METRLA			PEMSBAY		
	MAE	RMSE	WAPE	MAE	RMSE	WAPE
GATGPT	3.48	6.94	9.55%	1.84	4.07	4.14%
STLLM	3.44	6.89	10.04%	1.83	4.10	4.12%
LLM4ST	**3.34**	**6.88**	**9.37%**	**1.77**	**4.02**	**3.95%**

Table 4: Few-shot Experiments on METR-LA and PEMS-BAY.

LLM	PEMS04			PEMS08		
	MAE	RMSE	WAPE	MAE	RMSE	WAPE
GATGPT	24.81	37.62	22.18%	20.61	31.64	17.36%
STLLM	25.05	38.22	20.96%	20.67	31.36	19.50%
TPLLM	23.68	37.38	**15.57%**	18.09	28.51	**11.63%**
LLM4ST	**23.35**	**36.22**	17.97%	**17.94**	**28.17**	12.37%

Table 5: Few-shot Experiments on PEMS04 and PEMS08.

5.4 Few-shot Prediction

As shown in Table 4 and Table 5, LLM4ST-Traffic demonstrates significant advantages in few-shot scenarios. In the few-shot experiments on the METR-LA dataset (a scenario with high traffic flow fluctuations), its Weighted Average Percentage Error (WAPE) is 9.37%, which is 6.7% lower than that of STLLM (10.04%), proving that the model can still maintain prediction stability under extremely scarce data conditions. In the PEMS - BAY dataset (for short-term prediction tasks), the MAE of LLM4ST-Traffic is 1.77, which is 3.8% and 3.3% lower than that of GATGPT (1.84) and STLLM (1.83) respectively, indicating that it still performs excellently compared to other models when data is limited.

5.5 Visual Analysis of Semantic Alignment

To verify the alignment effect between traffic data and the semantic space, we conducted a visual analysis of the weights of the correlation matrices in the cross-attention mechanism, as shown in Figure 3. In this figure, the rows represent traffic data instances, the columns correspond to text words, and the color intensity reflects the strength of the association. To enhance the contrast, for each traffic data instance, we extracted the top 10 relevant words with the highest attention weights and the top 10 non-relevant words with the lowest attention weights for display. The results show that the highly relevant words include terms describing trends (such as "stable", "dropt", "+++", and

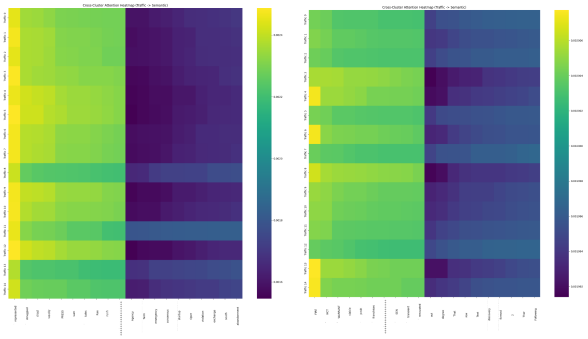

Figure 3: Cross-attention Maps in the Cross-modal Modules of PEMS-BAY (left) and METR-LA (right).

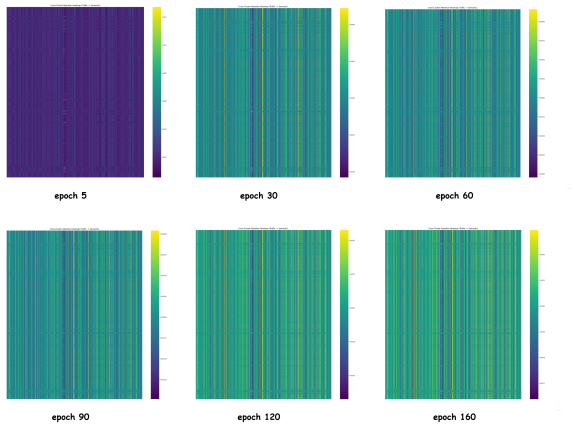

Figure 4: Cross-attention maps for different numbers of epochs

"peak") and periodic time-related words (such as "weekly"), indicating that the cross-modal module can effectively capture the correlation between the dynamic change patterns of traffic data and text semantics. Figure 4 shows the evolution of semantic alignment during the training process. In the initial training stage (epoch 5), the attention distribution is in a disordered state, and the correlation mapping is blurred. As the training progresses, the attention gradually focuses on domain-related words. The above visual results verify the model's ability to deeply align the spatio-temporal features of traffic data with text semantics.

6 Conclusion

This paper proposes LLM4ST-Traffic, a cross-modal traffic prediction framework based on Large Language Models, which addresses the challenge of data sparsity through semantic alignment and lightweight fine-tuning. The core contributions are: 1) the Cross-Modal Alignment (CMA) module, which dynamically associates spatiotemporal traffic features with textual semantics, overcoming the static nature and semantic disjunction of traditional linear mappings; and 2) Prefix Adapter Fine-Tuning (PAFT), which achieves a balance between performance and knowledge retention with minimal parameter adjustments. Experimental results demonstrate that this framework significantly outperforms mainstream methods across four benchmark datasets, excelling in low-sample scenarios, and includes an interpretability analysis. Future work will explore energy-efficient fine-tuning techniques to enhance the generalization ability of pre-trained models in downstream tasks.

7 Limitations

Although LLM4ST-Traffic demonstrates significant advantages in experiments, the following technical challenges still remain:

1. Information Loss in Cross-Modal Mapping: The current framework maps the entire traffic data into the discrete vocabulary space. Although it can utilize the text reasoning ability of the pre-trained model, the semantic-level representation has insufficient coverage of the high-order spatiotemporal patterns of traffic data (such as dynamic road network topology and the spread of unexpected events), limiting the model's fine-grained modeling ability for complex traffic scenarios.

2. Domain Adaptation Defects in Vocabulary Compression: Although the vocabulary compression strategy based on K-means (500 words) improves computational efficiency, only a part of the representative words generated are strongly related to the traffic scene, and the remaining words lack domain discrimination. This makes it difficult for the semantic alignment module to establish accurate traffic-text associations, and further exploration of domain knowledge-guided clustering optimization methods is required.

3. Dimension Conflict Caused by Spatiotemporal Structure Collapse: To adapt to the sequence input paradigm of the language model, it is necessary to compress the high-dimensional spatiotemporal structure of traffic data (batch × node × time step × dimension) into a three-dimensional sequence (batch × sequence × feature). This process destroys the local dependence of the spatio-temporal

topology and may weaken the model's ability to model the spatial propagation effect.

Future Improvement Directions: For limitation 1, we can consider using the pre-trained LLM as an external knowledge base, merely as a supplement to data sparsity rather than relying entirely on the LLM for feature extraction. For limitation 2, a customized compression strategy can be adopted. Some rules can be set in advance to make the vocabulary tend to generate words strongly related to the traffic scene when clustering. For limitation 3, contrastive learning can be designed. During the LLM tuning process, contrastive learning with traditional language models can be carried out to reduce the loss of traffic features.

References

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. 2023. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*.

Yakun Chen, Xianzhi Wang, and Guandong Xu. 2023. Gatgpt: A pre-trained large language model with graph attention network for spatiotemporal imputation. *arXiv preprint arXiv:2311.14332*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Yue Cui, Jiandong Xie, and Kai Zheng. 2021. Historical inertia: A neglected but powerful baseline for long sequence time-series forecasting. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 2965–2969.

Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W Tsang. 2021. St-norm: Spatial and temporal normalization for multi-variate time series forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 269–278.

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

AiLing Ding, XiangMo Zhao, and LiCheng Jiao. 2002. Traffic flow time series prediction based on statistics learning theory. In *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems*, pages 727–730. IEEE.

Frank Emmert-Streib, Zhen Yang, Han Feng, Shailesh Tripathi, and Matthias Dehmer. 2020. An introductory review of deep learning for prediction models with big data. *Frontiers in Artificial Intelligence*, 3:4.

Alex Graves and Alex Graves. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 922–929.

Mohammad M Hamed, Hashem R Al-Masaeid, and Zahi M Bani Said. 1995. Short-term prediction of traffic volume in urban arterials. *Journal of Transportation Engineering*, 121(3):249–254.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. 2023. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 4365–4373.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao. 2024a. Spatial-temporal large language model for traffic prediction. *arXiv preprint arXiv:2401.10134*.

Peiyuan Liu, H Guo, T Dai, N Li, J Bao, X Ren, Y Jiang, and ST Xia. 2024b. Calf: Aligning llms for time series forecasting via cross-modal finetuning. *arXiv preprint arXiv:2403.07300*.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. 2022. Frozen pretrained transformers as universal computation engines. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 7628–7636.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Yilong Ren, Yue Chen, Shuai Liu, Boyue Wang, Haiyang Yu, and Zhiyong Cui. 2024. Tpllm: A traffic prediction framework based on pretrained large language models. *arXiv preprint arXiv:2403.02221*.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 914–921.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*.

Hao Xue and Flora D Salim. 2023. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.

Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K Gupta, and Jingbo Shang. 2024. Large language models for time series: A survey. *arXiv preprint arXiv:2402.01801*.

Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE transactions on intelligent transportation systems*, 21(9):3848–3858.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1234–1241.

Zuduo Zheng and Dongcai Su. 2014. Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm. *Transportation Research Part C: Emerging Technologies*, 43:143–157.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355.

A Performance Analysis

The main observations are as follows:

- Performance Advantages across Multiple Datasets: As shown in Table 2 and Table 3, LLM4ST-Traffic significantly outperforms the baseline models in all prediction tasks (15/30/60 minutes) on the METR-LA and PEMS-BAY datasets. On the PEMS04 and PEMS08 datasets, its Mean Absolute Error (MAE) reaches 18.49 and 14.09 respectively, which is on average 7.4% lower than that of traditional spatio-temporal models (such as STGCN, AGCRN). This advantage stems from the efficient feature mapping ability of the cross-modal semantic alignment mechanism (CMA) and the design of the prefix-adapted fine-tuning strategy (PAFT). This enables it to not only lead in comparisons with traditional models but also maintain the optimal performance among LLM-integrated models.

- Comparative Advantages over LLM-Integrated Models: LLM4ST-Traffic reduces the average MAE by 12% compared to GATGPT and STLLM on the four datasets. Among them, the Maximum Absolute Percentage Error (MAPE) metric has a maximum improvement of 9.3% in the 15-minute prediction task on the PEMS-BAY dataset. In the comparison with LLM baselines, it improves the MAE by 5.3% and 8.8% on PEMS04/PEMS08 compared to TPLLM. The performance gap is due to the fact that the static linear projections relied on by GATGPT/STLLM make it difficult to achieve semantic-level alignment, resulting in weak associations between traffic patterns and text concepts. Thus, the capabilities of the pre-trained LLM cannot be fully exploited.

- Breakthrough in the Efficiency of Traditional Attention Models: Compared with attention-mechanism models, LLM4ST-Traffic demonstrates significant advantages in short-term prediction tasks: the MAE in 15-minute predictions is on average reduced by 5.71% (GMAN: 2.80 → 2.64). In complex scenarios (such as high traffic flow fluctuations in PEMS08), its MAE is significantly reduced by 23.13% compared to ASTGCN (18.33 → 14.09), verifying its strong adaptability to unexpected events.

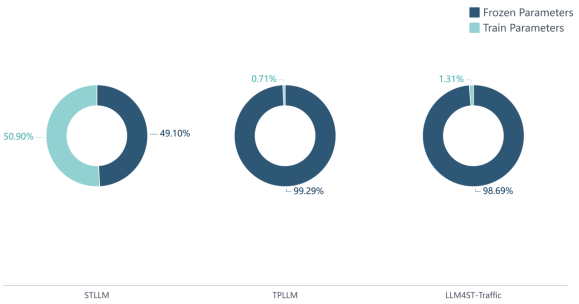

Figure 5: The proportion of LLM's own training parameters under different fine-tuning strategies.

Experiments show that LLM-based methods demonstrate significant advantages in traffic prediction tasks through open-domain knowledge transfer. LLM4ST-Traffic comprehensively surpasses existing models (including traditional spatiotemporal models and LLM-integrated methods) in the four benchmarks through semantic-driven alignment and lightweight knowledge transfer, providing an efficient solution for data-sparse scenarios. Its performance advantages and scalability mark a technological breakthrough of LLM in the field of spatio-temporal prediction.

B Different Fine-Tuning Strategies

Regarding the prediction effects under different fine-tuning schemes, we selected three schemes for comparison, namely the FPA method in STLLM, the LoRA method in TPLLM, and the Full Freeze method, as shown in Table 6. The effects of different schemes were verified on the METR-LA and PEMS-BAY datasets. The results indicate that LLM4ST-Traffic outperforms other models in all evaluation metrics (including MAE, RMSE, and WAPE), proving that the fine-tuning strategy we designed can effectively enhance the model performance.

In terms of the computational cost of fine-tuning, as shown in Figure 5, we compared the FPA method in STLLM with the LoRA method in TPLLM. On the premise of only considering the parameters of the LLM itself, the number of our trainable parameters is much lower than that of the FPA fine-tuning strategy in STLLM. The number of trainable parameters of our fine-tuning strategy is similar to that of the LoRA fine-tuning strategy, but our effect is better than that of the LoRA fine-tuning. This is

Datasets	Full Freeze			LORA			PFA			LLM4-Traffic		
	MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE
METR-LA	3.10	6.25	8.34	3.14	6.39	9.01	3.06	6.20	8.49	**2.95**	**6.06**	**7.96%**
PEMS-BAY	1.61	3.66	3.54	1.57	3.69	3.54	1.60	3.67	3.59	**1.55**	**3.62**	**3.44%**

Table 6: Performance comparison of different methods on METR-LA and PEMS-BAY.

because we designed a learnable prefix lightweight adaptation module. By adding an additional prefix prompt, the adaptability of the LLM to traffic tasks is enhanced. Meanwhile, only a very small number of LLM parameters need to be trained, which greatly reduces the computational cost of model training, maximally preserves the general prior knowledge of the LLM, and avoids the problem of knowledge forgetting.