Transportation Research Part C 191 (2026) 105851

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

A bicycle traffic prediction framework using pretrained large language models

Xiamei Wen *, Dorine Duives, Serge Hoogendoorn

Department of Transport and Planning, Delft University of Technology, Stevinweg 1, 2628CN, Delft, the Netherlands

ARTICLE INFO

Keywords:
Bicycle traffic prediction
Large language models
Spatial-temporal data

ABSTRACT

Bicycle traffic, as a sustainable mode of transport, helps reduce urban emissions and promote livable cities. Accurate short-term prediction of bicycle traffic flow can support real-time and near-real-time traffic management decisions in urban cycling systems. However, this task is challenging due to limited high-quality bicycle data, sparse sensor coverage, and high sensitivity to external factors. In this work, we propose a Bicycle Spatial-Temporal Large Language Model (BiSTLLM) for bicycle traffic prediction, inspired by the strong time series modeling capabilities of large language models (LLMs). In BiSTLLM, we define the historical traffic sequence at each location as tokens, allowing the model to learn complex spatial-temporal patterns. Specifically, we first introduce a weather lagging processor using causal temporal convolution to capture the delayed effects of weather on bicycle traffic. We also incorporate semantic embeddings of Point-of-Interest (POIs) and bicycle lane accessibility to represent land use and infrastructure, enhancing the model's ability to learn both localized activity patterns and broader spatial-temporal dynamics. We then further introduce a fine-tuning strategy to better tailor the general knowledge of a pre-trained LLM to the specific characteristics of bicycle traffic. Experiments on real-world bicycle traffic flow datasets demonstrate that BiSTLLM outperforms existing state-of-the-art models. Notably, BiSTLLM exhibits strong performance even in few-shot settings, highlighting its potential for accurate bicycle traffic prediction in data-sparse environments.

1. Introduction

The precise prediction of bicycle traffic flow is beneficial for efficient traffic management and potentially contributes significantly to improving service quality in a wide range of traffic-related applications that impact daily life (Yin et al., 2021). For example, a 10–30 min prediction can help adaptive signal systems prioritize cyclist phases or adjust clearance times during expected surges. In situations such as sports events, construction work, or temporary road closures, short-term predictions at key corridors can support anticipatory operational adjustments. Furthermore, bicycle traffic prediction can support operational planning and resource allocation during adverse weather conditions. Since bicycle monitoring systems are typically much sparser than motor vehicle sensing networks, developing robust prediction models under limited-data conditions is particularly important for practical real-world deployment. The traffic state within a road network is influenced not only by recent historical traffic flow at a specific location but also by the upstream and adjacent segments due to spatial connectivity within the network. Accurate traffic prediction, therefore, requires models capable of capturing these complex spatial and temporal dependencies. In recent years, deep learning models, which are a class of neural networks composed of more than one hidden layer, organized in deep nested network architectures (Janiesch et al., 2021; Yao et al.,

* Corresponding author.
E-mail address: X.Wen@tudelft.nl (X. Wen).

https://doi.org/10.1016/j.trc.2026.105851
Received 22 July 2025; Received in revised form 25 June 2026; Accepted 1 July 2026
Available online 6 July 2026

2025; Zhao et al., 2022; Ji et al., 2025a,b), have gained prominence in the domain of traffic prediction due to their ability to effectively model intricate traffic patterns across the network.

However, the accuracy of deep learning models is highly dependent on being trained with large volumes of high-quality traffic data (Yao et al., 2019). In the case of motorized traffic, such data is abundantly collected by companies and governments (Naghib et al., 2023). However, for active mode traffic, the availability of large-scale datasets is very limited, while shared-bike systems collect substantial demand data due to their investment returns. One key reason is that private bicycle traffic, unlike motorized traffic, is not typically associated with severe congestion issues. Consequently, it is often deprioritized in terms of resource allocation for data collection. The data sparsity undermines the ability of prediction models to capture network-wide influences, reducing their utility for traffic management and related applications (Liu et al., 2020). As a result, cyclists face greater uncertainty in their journeys, particularly during peak hours, as reliable information about traffic conditions and potential delays is unavailable. This limitation is especially pronounced in cycling-dominant countries like the Netherlands (Ton et al., 2019).

To achieve reliable traffic flow prediction, especially in essential monitoring points lacking sensor coverage or where sensors operate intermittently, pre-trained LLMs offer a transformative solution by leveraging their ability to infer patterns from incomplete data, integrate multimodal contextual signals (Zollner et al., 2024), and adapt to unseen scenarios through few-shot learning (Song et al., 2023). Although LLMs were originally developed for natural language processing tasks, such as text generation (Ramezani and Xu, 2023), translation (Xu et al., 2024), and semantic reasoning (Li et al., 2024c), their core architecture (self-attention mechanisms) (Vaswani et al., 2017) and pre-training paradigm (transfer learning) (Torrey and Shavlik, 2010) have proven equally powerful for time-series data modeling (Jin et al., 2023).

Recent studies have adapted LLMs for motorized traffic prediction, achieving good performance in tasks such as highway flow prediction (Lee and Gu, 2024), and urban demand prediction (Qu et al., 2024). Some efforts have also explored the application of LLMs to shared bicycle traffic prediction by incorporating features such as business Points of Interest (Li et al., 2024b), special event days, and weather conditions (Huang, 2024). However, these approaches remain underdeveloped for private bicycle traffic. Private bicycle traffic exhibits unique behavioral and environmental sensitivities that existing models often fail to capture. In particular, prior studies tend to overlook critical multimodal factors, such as the lagged effects of weather on bicycle flow (Wen et al., 2025b), where meteorological conditions influence bicycle demand, ridership patterns, route choices, and even operational needs minutes or even hours later, as well as geographic land use properties and the accessibility of nearby bicycle lanes.

To fully support effective traffic prediction and management strategies, this study proposes a Bicycle Traffic Prediction Framework Based on Pre-trained LLMs (BiSTLLM) for accurate bicycle traffic flow prediction, particularly in sensor networks with essential monitoring points that lack coverage or experience intermittent sensor activity. The proposed framework integrates spatial-temporal embeddings with pre-trained LLMs by leveraging its strong representation learning and transfer capabilities to enhance predictive performance in complex and data-sparse environments. In addition to the spatial-temporal representation of bicycle flow, BiSTLLM incorporates weather embeddings that capture the lagged effects of meteorological conditions on cycling behavior, geographic semantic embeddings derived from Points of Interest (POIs) to reflect location-specific activity patterns, and bicycle lane accessibility embeddings that model the semantics of cycling infrastructure and connectivity. These multi-modal contextual features enable the model to generalize effectively across diverse urban settings, even in the presence of limited sensor data. The main contributions of this paper are summarized as follows:

- We propose a Bicycle Traffic Prediction Framework Based on Pre-trained LLMs (BiSTLLM) for accurate bicycle traffic flow prediction. To enable LLMs to effectively interpret and model the dynamics of bicycle traffic, we develop a dedicated spatial-temporal embedding module that integrates one-dimensional convolution (1Dconv), Graph Convolutional Networks (GCNs), and a domain-specific external contextual feature processing module tailored to bicycle traffic. This design allows the model to capture and fuse rich spatial and temporal dependencies, as well as contextual semantics, that are intrinsic to bicycle traffic patterns.
- To address the delayed effects of weather on bicycle traffic, we design a lagging processor that dynamically fuses weather embedding with spatial-temporal bicycle traffic flow embedding. Specifically, we employ a flow-gated mechanism to adaptively scale the influence of weather conditions based on the current traffic flow. Subsequently, a Causal Temporal Convolution (CausalConv) is applied to model the time-lagged impact of weather on traffic dynamics effectively.
- To evaluate the practical applicability of the proposed approach in realistic bicycle traffic scenarios, we conduct experiments using data from multiple cities under both high-data and low-data conditions. Extensive experimental results demonstrate that BiSTLLM consistently outperforms a range of state-of-the-art baseline models.

The remainder of this paper is structured as follows. Section 2 reviews related work on spatial-temporal traffic prediction and the application of pre-trained LLMs to time series prediction. Section 3 formulates the problem addressed in this study. Section 4 details the architecture and components of the proposed BiSTLLM framework. Section 5 describes the experimental setup and presents the empirical results. Finally, Section 6 discusses the broader implications and key contributions of our approach.

2. Related work

2.1. Data-driven spatial-temporal traffic prediction

Traditional traffic prediction approaches, such as historical average (HA) and time-series statistical prediction model like AutoRegressive Integrated Moving Average (ARIMA) (Box and Pierce, 1970) have shown limited effectiveness in capturing the complex, nonlinear spatial-temporal dependencies inherent in traffic data. To address some of these limitations, machine learning models, including Support Vector Regression (SVR) (Awad et al., 2015) and Random Forests (Breiman, 2001), were introduced. However, these approaches primarily focused on modeling nonlinear temporal patterns while neglecting spatial dependencies. With the advent of deep learning, models such as Long Short-Term Memory (LSTM) networks (Graves and Graves, 2012), Gated Recurrent Unit (GRU) (Cho et al., 2014) have been widely adopted to learn temporal dynamics. Convolutional Neural Networks (CNNs) (O'shea and Nash, 2015) have been used to model local spatial and temporal correlations through convolutional operations, while Graph Neural Networks (GNNs) (Wu et al., 2020) have been leveraged to capture spatial relationships. These deep learning models align closely with the intrinsic characteristics of traffic flow, offering significant improvements in the accuracy and robustness of spatial-temporal traffic prediction.

More recent hybrid models integrate deep learning architectures to jointly capture spatial and temporal features, further enhancing the performance of traffic prediction systems. For example, a novel Attention-based Periodic-Temporal neural Network (APTN) is proposed by Shi et al. (2020) as an end-to-end solution for traffic prediction, designed to capture spatial, short-term, and long-term periodic dependencies. Several studies have further incorporated external factors such as weather, events, and road conditions to enrich spatial-temporal representations. For instance, Qi et al. (2024) proposed a spatial-temporal fusion graph convolutional network (STFGCN) that leverages multi-feature fusion to explicitly account for weather conditions, demonstrating improved prediction performance over representative baselines. Similarly, Ye et al. (2022) proposed an Attention-Based Spatio-Temporal Graph Convolutional Network considering External Factors (ABSTGCN-EF), which models contextual features such as time of day, day of the week, and traffic accident events. Their results show that incorporating external information can significantly enhance predictive accuracy. While these approaches improve representation learning by integrating contextual features, they generally rely on dense and continuous sensor data to achieve strong performance. However, in practice many bicycle traffic monitoring locations suffer from insufficient sensor coverage or intermittent operation, resulting in sparse and incomplete historical records. Under such data-limited conditions, conventional deep learning models often struggle to generalize effectively. This limitation highlights the need for models with stronger transferability and robustness in low-data scenarios.

2.2. *Pre-trained LLMs for time-series prediction*

Recent advancements in pre-trained LLMs have significantly expanded their applications beyond natural language processing into time-series prediction tasks (Jiang et al., 2024). This cross-domain transferability stems from their architecture of modeling sequential dependencies through self-attention mechanisms (Vaswani et al., 2017). Recently, LLMs have demonstrated remarkable potential in capturing both short-term fluctuations and long-term trends in complex time-series data in domains such as finance, healthcare, and energy systems (Ye et al., 2024). For example, Yu et al. (2023) present a novel study that harnesses the knowledge and reasoning capabilities of Large Language Models (LLMs) for explainable financial time-series forecasting. Li et al. (2024a) propose Multimodal ECG-Text Self-supervised pre-training (METS) to improve electrocardiogram (ECG) data classification, enabling zero-shot prediction and improving model generalization without relying on annotated data.

In the domain of traffic prediction, pre-trained LLMs have recently been explored as a promising alternative to conventional spatial-temporal traffic prediction models, due to their ability to exploit structural similarities between traffic data and language, such as sequences and context, and to adapt through fine-tuning to capture complex temporal dependencies. For example, Liu et al. (2024) proposed a Spatial-Temporal Large Language Model (ST-LLM) for traffic prediction based on a partially frozen attention strategy to adapt the LLM to capture global spatial-temporal dependencies for traffic prediction. Experiments on the NYCTaxi and CHBike datasets offer evidence that ST-LLM is a powerful spatial-temporal learner that outperforms state-of-the-art models. Ren et al. (2024) introduced a novel traffic prediction framework leveraging LLMs (TPLLM). A sequence embedding layer based on Convolutional Neural Networks (CNNs) and a graph embedding layer based on Graph Convolutional Networks (GCNs) is used to extract sequence features and spatial features. Experiments on PeMS04 and PeMS08 datasets collected by California Department of Transportation demonstrate that TPLLM exhibits commendable performance in both full-sample and few-shot prediction scenarios. Xu and Liu (2025) propose a Spatio-temporal Fusion Large Language model (GPT4TFP) for traffic flow prediction, which is divided into four components: the spatio-temporal embedding layer, the spatio-temporal fusion layer, the frozen pre-trained LLM layer, and the output linear layer. The experimental results on NYCTaxi, CitiBike, PEMS04 and PEMS08 traffic flow datasets show that the proposed model outperforms a set of state-of-the-art baseline models. Current applications of LLMs in traffic prediction have predominantly focused on motorized vehicles and shared-bike systems. However, private bicycle traffic prediction remains relatively underexplored despite its unique challenges. Private bicycle flows exhibit greater behavioral variability (Wen et al., 2025a). Moreover, the influence of environmental factors and localized infrastructure on private bicycle usage has yet to be thoroughly investigated.

3. Problem formulation

The ability to reliably predict traffic conditions at locations with limited data collection is particularly crucial for infrastructure planning, real-time mobility management, and safety enhancement in bicycle mobility and traffic flow, as accurate prediction in these locations could support proactive traffic monitoring and operational decision-making by providing early indications of expected bicycle volumes. To address these challenges, this study develops a novel prediction framework that is capable of providing an accurate and reliable prediction for bicycle traffic, even in cases where historical data from some sensors is limited.

Definition (Active-mode traffic network): Given the challenges associated with collecting bidirectional bicycle traffic data, we define the bicycle traffic network as an undirected graph $\mathcal{G} = (V, E, \mathbf{A})$, where V represents the set of nodes corresponding to sensors that monitor bicycle traffic flow within the road network; E denotes the set of edges capturing the connectivity between these sensors; and $\mathbf{A} \in \mathbb{R}^{\mathbb{N}\times\mathbb{N}}$ is the weighted adjacency matrix of $\mathcal{G}$ encoding the structural relationships among the N nodes.

Problem (Multi-step short-term traffic prediction): The input for traffic prediction consists of sensor traffic flow data from the past hour. For each sensor, the traffic flow data is represented as $\mathbf{X} = (\mathbf{X}_{t-T_k+1}, \mathbf{X}_{t-T_k+2}, \dots, \mathbf{X}_t) \in \mathbb{R}^{N\times F\times T_k}$, where T_k denotes the number of time steps in each input sequence. In addition, the historical data at each location is treated as a token. At each time step t, the feature matrix is given by $\mathbf{X}_t = (\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \dots, \mathbf{x}_{t,N}) \in \mathbb{R}^{N\times F}$, where N is the number of sensors, and F represents the feature dimension of each node.

Our goal is to find a function f to predict the following T time steps data $\hat{\mathbf{y}} = (\hat{\mathbf{y}}_{t+1}, \hat{\mathbf{y}}_{t+2}, \dots, \hat{\mathbf{y}}_{t+T}) \in \mathbb{R}^{N\times F\times T}$, that is:

$$(\hat{\mathbf{y}}_{t+1}, \hat{\mathbf{y}}_{t+2}, \dots, \hat{\mathbf{y}}_{t+T}) = f_\theta((\mathbf{X}_{t-T_k+1}, \dots, \mathbf{X}_t), \mathbf{A}_t) \tag{1}$$

Here, θ represents the learnable parameters of the function. In this paper, we used the previous one hour of data to predict traffic flow for the next hour.

4. The BiSTLLM framework

This section presents the details of the proposed BiSTLLM. As illustrated in Fig. 1, the framework employs a pre-trained LLM as the core prediction engine, augmented with a customized architecture designed to capture domain-specific spatial-temporal embeddings for bicycle traffic flow prediction. BiSTLLM comprises three primary components: (1) extraction of spatial-temporal bicycle traffic flow embeddings, (2) modeling of bicycle traffic-specific external contextual embeddings, and (3) LLM-based prediction utilizing enriched spatial-temporal embedding.

The framework begins by processing the bicycle traffic data through parallel modules for spatial-temporal embeddings capturing:

- **Temporal embedding capturing:** 1DConv (Bai et al., 2018) is used to encode sequential patterns in historical bicycle flow data, effectively capturing local temporal dependencies.
- **Spatial embedding capturing:** The GCN generates node embeddings that encode structural dependencies within the bicycle traffic network.

To enhance the representational capacity of the model and capture the unique characteristics of bicycle traffic, the following bicycle traffic-specific external contextual feature embedding are incorporated:

- **Weather condition embedding capturing:** Meteorological data undergoes temporal alignment through a weather lagging processor that accounts for delayed weather impacts on bicycle flows.
- **Geographic context embedding capturing:** Contextual features derived from POIs and bicycle lane accessibility are encoded through prompt-based semantic extraction using the pre-trained LLM, capturing geographic and infrastructural factors that influence cycling behavior.

These embeddings are then fused and organized into a unified input stream for the LLM:

- **Input embeddings construction:** The spatial-temporal traffic flow embeddings are fused with temporal indicators, processed weather embedding, geographic semantic context, and node-specific embeddings to form a rich, structured representation.

The pre-trained LLM processes these inputs based on the fine-tuning strategy of Section 4.4, enabling the model to jointly capture complex spatial and temporal dependencies. This design facilitates accurate, context-aware predictions of bicycle traffic flow across diverse urban environments.

4.1. Spatial-temporal bicycle traffic flow embeddings

4.1.1. Temporal dependency capture module

While LLMs are powerful sequence learners, they are primarily optimized for processing natural language and may lack the inductive biases necessary to effectively capture short-term, domain-specific temporal patterns critical for traffic flow prediction. To address this limitation, we introduce a series of 1DConv layers as a preprocessing module prior to inputting the data into the pre-trained LLM. This design choice is particularly relevant for traffic sequences, where temporal fluctuations are often influenced by dynamic external factors such as weather conditions, special events, and variations in travel demand behavior. The 1DConv layers explicitly encode these fine-grained temporal features, enhancing the quality and informativeness of the input representations. By capturing temporal dependencies of bicycle traffic before feeding data into the pretrained LLM, the model may be better equipped to learn complex temporal patterns, potentially improving accuracy and robustness in traffic flow predictions.

$$\mathbf{T_e} = \mathrm{1DConv}(\mathbf{X}; \mathbf{W}, \mathbf{b}) \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{k\times C\times C'}$ denotes the convolutional kernel weights with a kernel size of 3, mapping from C input channels to C' output channels; $\mathbf{b} \in \mathbb{R}^{C'}$ is the bias term; $\mathbf{T_e} \in \mathbb{R}^{T'\times C'}$ is the output temporal embedding; T' is the output sequence length; $\mathbf{X}$ is the input traffic flow data, which includes only feature flow.

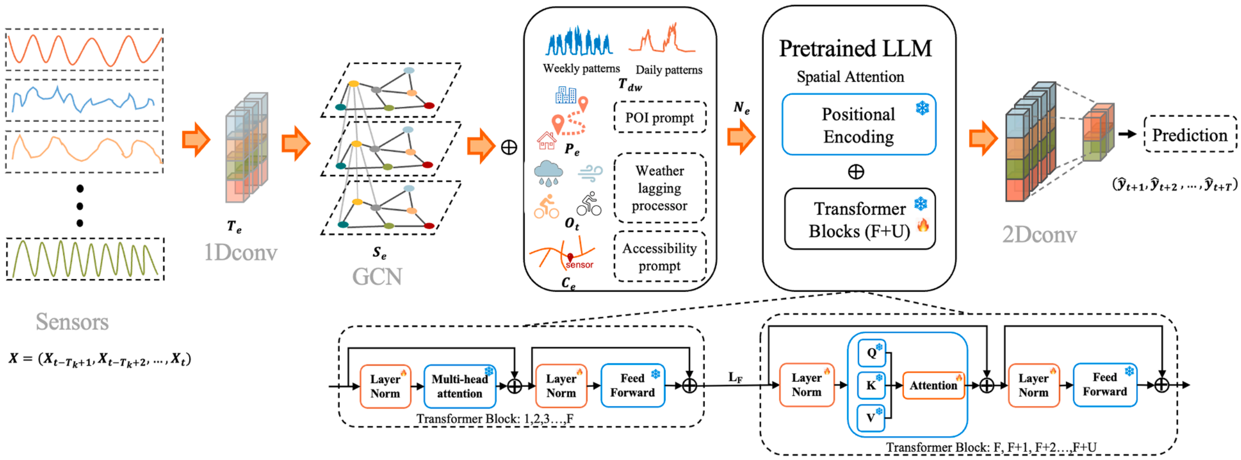

Fig. 1. The Framework of BiSTLLM. Here, Q, K, and V refer to the Query, Key, and Value representations in multi-head self-attention, respectively. In our case, they are traffic embeddings, obtained by applying learned linear transformations to the input traffic data.

4.1.2. Spatial dependency capture module

The bicycle traffic network is modelled as a graph, where nodes represent sensors, and edges capture their spatial correlations. Each node is associated with traffic features derived from historical traffic flow data. Spatial-based GCN (Kipf and Welling, 2016) performs successive graph convolutions using an adjacency matrix $\mathbf{A}$, allowing the model to aggregate information from neighboring nodes. This enables the network to learn feature representations that encapsulate local and global spatial dependencies within the traffic network. Through this process, spatial-based GCN effectively refines the feature representations of each node, facilitating improved downstream pre-trained LLM prediction performance. The corresponding operational Eq. (3) is formulated as follows:

$$\mathbf{S_e} = ReLU(\mathbf{A}\mathbf{X}_t\mathbf{W}) \tag{3}$$

where the trainable weight matrix, denoted by $\mathbf{W}$ is used to transform the node features during the graph convolution operation. The Rectified Linear Unit $ReLU(\cdot)$ is employed as the activation function to introduce non-linearity into the model, ensuring its capacity to capture complex patterns in the data. $\mathbf{X}_t$ is the output from temporal dependency capture module.

The sensors distributed across the road network are modeled as nodes within a graph, where the Dijkstra algorithm (Ojekudo and Akpan, 2017) is employed to compute the shortest paths between nodes. These shortest paths define the topological connectivity of the network, capturing the structural relationships between sensors. Recognizing that traffic patterns at a given sensor are often influenced by the behavior of neighboring sensors, where closer sensors tend to exhibit more similar traffic patterns, a distance-based adjacency matrix is introduced to establish weighted connections between sensors. The weights in this matrix are derived using a Gaussian kernel weighting function (Shuman et al., 2013), as formulated in Eq. (4), which assigns higher weights to pairs of sensors that are geographically closer and lower weights to those that are farther apart. To further capture latent spatial correlations and enhance model adaptability, we treat this adjacency matrix as a learnable parameter. This allows the model to dynamically adjust spatial dependencies during training, thereby more accurately modeling the influence of neighboring sensors on local traffic flow patterns.

$$Dis_d^{i,j} = \begin{cases} \exp(-\frac{|\mathrm{dist}(i,j)|^2}{2\hat{\theta}^2}) & \text{if } dist(i,j) < H \\ 0 & \text{else} \end{cases} \tag{4}$$

where the weight of the graph's edges is denoted by $Dis_d^{i,j}$, $|\mathrm{dist}(i,j)|$ indicates the Euclidean distance between node i and node j. In this context, $\hat{\theta}$ represents the standard deviation of distances, and H is utilized as the threshold parameter.

4.2. Bicycle traffic-specific external contextual feature processing module

4.2.1. Weather lagging processor

Bicycle traffic is highly sensitive to external environmental factors, particularly weather conditions (Hanson and Hanson, 1977; Böcker et al., 2013; Wen et al., 2025a). For instance, bicycle flow typically decreases during periods of rainfall or strong winds. However, this reduction in traffic flow does not necessarily occur immediately with the onset of adverse weather; instead, a lagged effect is often observed between the change in weather and its impact on bicycle traffic (Wen et al., 2025b), as shown in Fig. 2. And the magnitude of the delayed response may vary substantially under different traffic states and weather conditions. Moreover, the influence of weather is nonuniform across different traffic volumes. For example, weather effect on low traffic volumes may differ from its impact during peak periods. To accurately model the influence of weather on bicycle traffic, The weather impacts is first dynamically scaled based on real-time traffic conditions using a sigmoid-based adjustment mechanism. Subsequently, we employ a

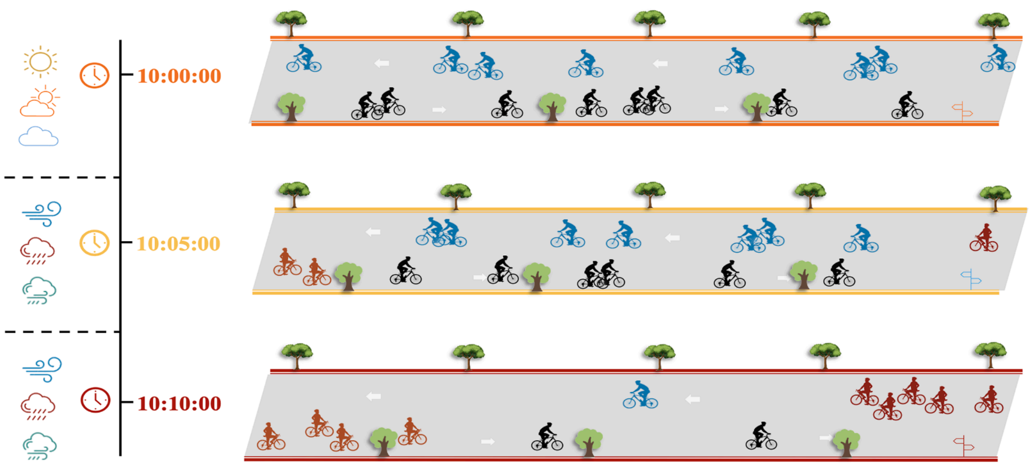

Fig. 2. The delayed effect of weather conditions on bicycle traffic patterns. The black and blue bicycle icons indicate different cycling directions. The orange and red bicycle icons represent cyclists exhibiting a lagged response to adverse weather conditions across different cycling directions: some cyclists stop and wait immediately, while others continue their trips despite unfavorable weather.

causal temporal convolution (Howard et al., 2017) to capture the delayed effects of weather on traffic flow with a kernel size of 10, as shown in Eq. (5).

$$\mathbf{o}_t = \text{DepthwiseConv1D}\big((\mathbf{W}_r\mathbf{p}_t + \mathbf{W}_w\mathbf{s}_t) \otimes \sigma(\mathbf{W}_f\mathbf{X}_t)\big) \tag{5}$$

where $\mathbf{o}_t \in \mathbb{R}^d$ is the output lagging weather representation at time t, $\mathbf{W}_r, \mathbf{W}_w, \mathbf{W}_f \in \mathbb{R}^{d\times 1}$ are learnable projection weights, $\mathbf{p}_t, \mathbf{s}_t, \mathbf{X}_t \in \mathbb{R}$ denote rainfall, wind speed, and bicycle traffic flow produced by the spatial dependency capture module respectively, $\sigma(\cdot)$ is the sigmoid activation function, $\otimes$ represents the Hadamard (element-wise) product, and DepthwiseConv1D$(\cdot)$ applies a causal depthwise convolution so that the convolutional filter never sees the future weather information.

4.2.2. Geographic feature embedding capture

POI-based semantic embedding generation: Traffic patterns within a road network are strongly influenced by the surrounding land use characteristics Izanloo et al. (2017). For example, sensors deployed in areas with a high density of restaurants often exhibit distinct flow patterns due to increased human activity, while zones dominated by educational institutions, such as schools, reflect different temporal behaviors. Regions that share similar land use characteristics, such as areas dense with restaurants or commercial establishments, may exhibit analogous traffic patterns, even if geographically distant. To capture these latent spatial correlations, we extract textual descriptions of POIs surrounding each sensor location using OpenStreetMap (OSM) data. These textual POI metadata include categories such as food services, retail, entertainment, public services, lodging, and cultural institutions. To ensure consistency and interpretability, we construct structured natural language descriptions using a predefined template. Each description summarizes the distribution of nearby POI categories for a given node. We then leverage a pre-trained LLM, GPT-2, as a semantic feature extractor to convert the POI textual metadata into dense vector embeddings. Specifically, the POI text associated with each sensor location is tokenized using GPT-2's tokenizer and fed through GPT-2's embedding layers to generate semantic-rich feature vectors $\mathbf{P}_e$, as defined in Eq. (6). These embeddings serve as auxiliary input embeddings, which are concatenated with traffic flow spatial-temporal embedding in the proposed BiSTLLM framework to enhance the model's understanding of spatial context.

$$\mathbf{P}_e = \mathcal{E}_{\text{GPT-2}}\big(\mathbf{T}_{\text{POI}}\big(P_{\text{id}}(\text{Lat}, \text{Lon})\big)\big) \tag{6}$$

where $P_{\text{id}}(\text{Lat}, \text{Lon})$ denotes the geographic coordinates of the sensor. $\mathbf{T}_{\text{POI}}(\cdot)$ denotes POI-related textual metadata; $\mathcal{E}_{\text{GPT-2}}(\cdot)$ indicates the embedding extraction operation using GPT-2.

Bicycle lane accessibility semantic embedding generation: The availability and connectivity of bicycle lanes significantly influence urban traffic dynamics, especially for regions with a high density of non-motorized mobility, and areas with extensive and well-connected bicycle lane infrastructure tend to become active cycling corridors, which in turn shape local traffic flow patterns Dill and Carr (2003). Furthermore, geographically distant regions that share similar levels of bicycle infrastructure accessibility may exhibit analogous traffic behaviors. To capture this form of structural spatial context, A bicycle lane accessibility prompt is designed, which encodes the degree of bicycle lane integration at each sensor location. Specifically, for each traffic sensor, we extract the sub-network of bicycle lanes based on OSM located within a 100-meter radius of its geographic position. Within this localized network, we identify the number of distinct bicycle lane segments directly connected to the topological lane on which the sensor resides. This connectivity count serves as a proxy for local infrastructure integration and accessibility. The resulting numerical accessibility value is then transformed into a descriptive textual prompt. This textual metadata is tokenized using GPT-2's tokenizer and subsequently passed through the embedding layer to obtain the semantic representation $\mathbf{C}_e$, as shown in Eq. (7):

$$\mathbf{C}_e = \mathcal{E}_{\text{GPT-2}}\big(\mathbf{T}_{\text{bike}}\big(\text{Net}_{\text{sensor}}(\text{Lat}, \text{Lon})\big)\big) \tag{7}$$

here, $\mathrm{Net}_{\mathrm{sensor}}(\mathrm{Lat}, \mathrm{Lon})$ denotes the local bicycle lane sub-network centered at the sensor's geographic location. $\mathbf{T}_{\mathrm{bike}}(\cdot)$ generates the lane-related textual metadata, and $\mathcal{E}_{\mathrm{GPT\text{-}2}}(\cdot)$ represents the GPT-2-based embedding function applied to the tokenized prompt.

4.3. Input embedding fusion

We concatenate the previously described spatial-temporal embeddings into a unified representation, which serves as the input to the pre-trained LLM. Before concatenation, each feature representation is first projected into a shared embedding space with dimension as d_{model}. To align these embeddings with the input structure expected by the LLM, we employ a two-dimensional convolutional neural network (2DConv) with a kernel size of (1,3) for formatting and transformation, which serves purely as a linear projection for feature fusion without applying any activation functions. This step ensures that spatial-temporal embeddings are not only effectively fused but also organized in a tokenized structure compatible with the LLM architecture. The 2DConv module captures local spatial and temporal dependencies across both dimensions, thereby enhancing the contextual coherence of the input sequence. This transformation allows the LLM to better interpret domain-specific traffic dynamics and facilitates end-to-end learning for downstream prediction tasks. The equation is shown below:

$$\mathbf{I}_e = 2Dconv(Concate(\mathbf{T}_{\mathbf{e}}, \mathbf{S}_e, \mathbf{T}_d w, \mathbf{N}_e, \mathbf{o}_t, \mathbf{C}_e, \mathbf{P}_e), \mathbf{W}_{2d}) \tag{8}$$

where $\mathbf{T}e$ denotes the temporal input embedding to the pre-trained LLM, and $\mathbf{S}e$ represents the spatial input embedding. $\mathbf{o}_t$ corresponds to the lagged weather embedding, while $\mathbf{T}_d w$ denotes the time-of-day and day-of-week embeddings, $\mathbf{P}_e$ represents the POI embedding, $\mathbf{C}_e$ is the bicycle lane accessibility embedding, and $\mathbf{N}_e$ corresponds to the node-specific adaptive embedding used to encode the traffic network structure (Liu et al., 2024). $\mathbf{W}_{2d}$ represents the learnable parameters of 2Dconv.

4.4. Pre-trained LLM fine-tuning strategy

Fine-tuning the pre-trained LLM is critical to facilitate downstream tasks based on domain-specific datasets, enabling the models to better capture task-relevant semantics and contextual nuances (VM et al., 2024). Prior approaches such as the Frozen Pretrained Transformer (FPT) (Lu et al., 2022) and the Partially Frozen Attention (PFA) LLM (Liu et al., 2024) have demonstrated strong performance across a variety of downstream tasks. However, short-term bicycle flow prediction presents unique challenges that may limit their effectiveness. To enhance the performance of pretrain-LLM for short-term bicycle traffic prediction, we introduce a fine-tuning strategy for the proposed BiSTLLM framework. Specifically, we freeze the multi-head attention (MHA) and feed-forward network (FFN) sublayers in the first F transformer blocks while allowing updates only to the layer normalization (LN) components. This design preserves the general linguistic and structural priors learned during pretraining while enabling modest task-specific adaptation. In the subsequent U transformer layers, we further allow updates to the attention mechanisms by unfreezing the attention computations and LN sublayers, while keeping the FFN and the query, key, and value projection matrices frozen. Additionally, we retain the positional encoding in a fixed state, as we focus on spatial dimensions in pre-trained LLM and positional encoding along this dimension is not semantically meaningful. This fine-tuning strategy balances adaptability and efficiency, allowing the model to capture domain-specific patterns in bicycle traffic flow data while minimizing the risk of catastrophic forgetting and reducing computational overhead. In contrast, randomly freezing ignores the functional roles of different modules and can lead to unstable training dynamics or degraded performance.

The fine-tuning approach adopted in this study is based on a pre-trained LLM constructed using the GPT-2 Transformer architecture with small size. GPT-2, as an autoregressive model built upon the Transformer decoder, offers a relatively compact parameter structure, making it well-suited for traffic flow prediction tasks due to its balance between expressiveness and computational efficiency (Zhou et al., 2023). The architectural modifications applied to different components of the model during fine-tuning are illustrated in the lower section of Fig. 1. The formulas are shown in Eqs. (9) and (10)

$$\begin{aligned} \bar{\mathbf{E}}_i &= \mathrm{MHA}(\mathrm{LN}(\mathbf{E}_i)) + \mathbf{E}_i \\ \mathbf{E}_{i+1} &= \mathrm{FFN}(\mathrm{LN}(\bar{\mathbf{E}}_i)) + \bar{\mathbf{E}}_i \end{aligned} \tag{9}$$

where $i \in 1, \dots, F-1$; $\mathbf{E}_1 = \mathbf{I}e + \mathbf{P}_E$, with $\mathbf{P}_E$ denoting the frozen positional encoding; $\mathbf{E}_i$ is the input to the ith Transformer block; $\bar{\mathbf{E}}_i$ represents the intermediate output after applying the unfrozen layer normalization $\mathrm{LN}(\cdot)$ followed by the frozen multi-head attention $\mathrm{MHA}(\cdot)$; and $\mathbf{E}_{i+1}$ is the final output after applying the unfrozen $\mathrm{LN}(\cdot)$ and the frozen feed-forward network (FFN) to $\bar{\mathbf{E}}_i$.

$$\begin{aligned} \bar{\mathbf{E}}_{F+U-1} &= \mathrm{MHA}(\mathrm{LN}(\mathbf{E}_{F+U-1})) + \mathbf{E}_{F+U-1} \\ \mathbf{E}_{F+U} &= \mathrm{FFN}(\mathrm{LN}(\bar{\mathbf{E}}_{F+U-1})) + \bar{\mathbf{E}}_{F+U-1} \end{aligned} \tag{10}$$

where $\bar{\mathbf{E}}_{F+U-1}$ denotes the intermediate representation at the $(F + U - 1)$th layer, obtained by applying the unfrozen layer normalization $\mathrm{LN}(\cdot)$ and the unfrozen multi-head attention $\mathrm{MHA}(\cdot)$ with frozen query, key, and value projections; and $\mathbf{E}_{F+U}$ represents the final output after applying the unfrozen $\mathrm{LN}(\cdot)$ and the frozen feed-forward network (FFN) to $\bar{\mathbf{E}}_{F+U-1}$.

Algorithm 1: The BiSTLLM Framework

Input: Bicycle traffic features $\mathbf{X}_t$ over T_k historical timesteps; training epochs E_r; learning rate η; batch size B_z; optimizer: Ranger21.
Output: Trained BiSTLLM model.
foreach *epoch* $r \in \{1, 2, \ldots, E_r\}$ **do**
  Shuffle the training data;
  foreach *batch* $\mathbf{X}_T$ *in training data* **do**
    $\mathbf{T_e}, \mathbf{S}_e, \mathbf{T}_d w, \mathbf{N}_e, \mathbf{o}_t, \mathbf{C}_e, \mathbf{P}_e \leftarrow$ Spatial and Temporal Embedding using Equations (2), (3), (5), (6), and (7);
    $\mathbf{I}_e \leftarrow$ Fuse embeddings using Equation (8);
    for $i = 1$ **to** $F + U$ **do**
      if $i \leq F$ **then**
        Compute $\mathbf{E}^{i+1}$ using Equation (9) with $\mathbf{E}^i$;
      else
        Compute final output $\mathbf{E}^{F+U}$ using Equation (10) with $\mathbf{E}^{F+U-1}$;
      end
    end
    $\hat{y} \leftarrow$ The prediction output;
    Update all learnable parameters by minimizing the loss in Equation (11) via Ranger21 optimizer.
  end
end

4.5. Loss function

Due to the presence of noise and outliers in bicycle traffic flow data caused by sensor failures or environmental interference, we adopt the robust L1 loss function in BiSTLLM. L1 loss is less sensitive to extreme deviations, making it more suitable for handling anomalies in real-world traffic datasets.

$$\mathcal{L}_{\text{loss}} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{11}$$

here, N is the total number of samples; y_i denotes the ground truth value of the ith sample; and $\hat{y}_i$ is the corresponding predicted value. The overall workflow of BiSTLLM is outlined in Algorithm 1.

5. Experimental results and analysis

In this section, we first describe the datasets and evaluation metrics employed in our experiments. We then outline the baseline models and detail the experimental settings used for performance comparison. Finally, we present and analyze the prediction results, followed by an ablation study to evaluate the contribution of each component in the proposed framework.

5.1. Dataset analysis

To assess the effectiveness of the proposed model, we conducted extensive experiments on real-world bicycle traffic flow data obtained from the National Road Traffic Data Portal of the Netherlands. The dataset as shown in Table 1, comprises measurements recorded by loop detector sensors strategically deployed across two Dutch cities: Rotterdam and Leiden. For Rotterdam, the dataset spans from September 1, 2024, to November 30, 2024, covering 18 sensor locations. Similarly, for Leiden, data were collected from 18 sensors over the period from January 1, 2025, to April 30, 2025. The sensors distribution are shown in Fig. 3. The raw sensor readings were preprocessed by aggregating the traffic flow data into 5 min intervals, resulting in 12 observations per hour. This temporal resolution offers a practical balance between granularity and computational efficiency. To facilitate efficient model training and improve convergence, the aggregated data were standardized using z-score normalization, ensuring zero mean and unit variance across features. For a rigorous and unbiased evaluation, the dataset was chronologically partitioned into training, validation, and test sets using a 6:2:2 split. This forward-looking split methodology mitigates data leakage and ensures that the model is evaluated on future, unseen observations, thereby providing a realistic assessment of its predictive performance and generalization capability.

Weather information was obtained from the Royal Netherlands Meteorological Institute (KNMI), specifically from the Rotterdam and Voorschoten weather stations. The dataset includes hourly observations of key meteorological variables, namely precipitation (measured in millimeters) and mean wind speed (measured in 0.1 m/s). Precipitation represents the total rainfall recorded in the preceding hour, while wind speed reflects the average wind conditions over the same period. To align the weather data temporally with the 5 min bicycle traffic flow intervals, each hourly weather observation was divided into 12 equal sub-intervals. The mean wind speed was replicated across all 12 sub-intervals, assuming consistent wind conditions within each hour. For precipitation, the hourly value was uniformly distributed across the 12 intervals to approximate the average rainfall per 5 min period. This interpolation strategy allows for fine-grained temporal alignment between meteorological data and traffic flow observations, thereby enhancing the model's ability to capture lagged or cumulative weather effects on cycling behavior.

Table 1
Datasets information.

Datasets	Time Span	Number of Sensors	Aggregation Granularity
Rotterdam	2024/09/01-2024/11/30	18	5-min
Leiden	2025/01/01-2025/04/30	18	5-min

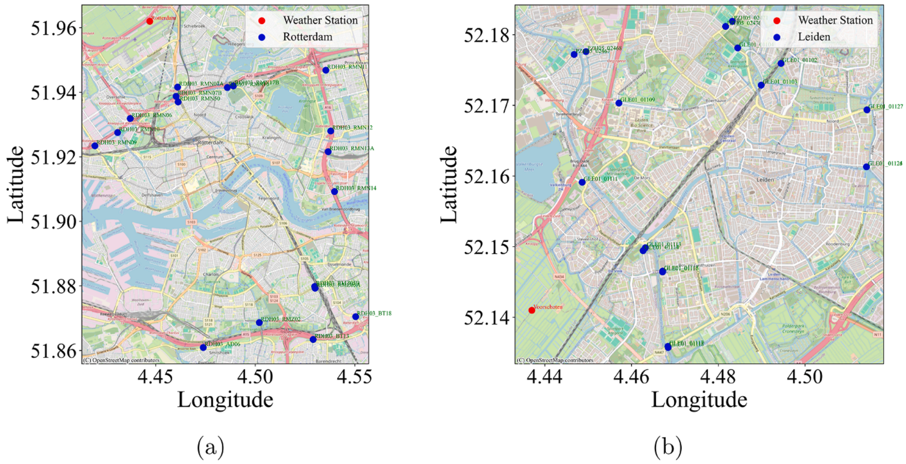

Fig. 3. Sensors and weather station distribution. (a) Rotterdam and Rotterdam weather station. (b) Leiden and Voorschoten weather station.

In this paper, we use traffic flow data from the previous hour (12 observations per hour) to predict traffic flow for the next hour at 5 min intervals.

5.2. Evaluation metrics

In this paper, the prediction results of bicycle traffic flow are evaluated by Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Weighted Absolute Percentage Error (WAPE). The formulations to calculate these metrics are shown below.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{12}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{13}$$

$$WAPE = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{\sum_{i=1}^{n}|y_i|}, \tag{14}$$

where $\hat{y}_i$ and y_i represent the predicted value and ground truth data of the sample i, respectively. n is the number of sample values.

5.3. Baseline methods

To assess the performance of our proposed BiSTLLM model in predicting bicycle traffic flow, we compare it with several previously used traffic prediction models. Through this comparison, our aim is to highlight BiSTLLM's capability in predicting bicycle traffic. The baseline models used for comparison are briefly described below:

- HA: The Historical Average (HA) model predicts traffic flow by taking the average value of historical data. Specifically, the HA baseline is computed as the per-sensor, per historical observations average.
- LSTM: Long Short-Term Memory (LSTM) (Graves and Graves, 2012) is a specialized variant of Recurrent Neural Networks (RNNs) specifically designed to effectively capture and model long-term relations in sequential data.
- STGCN: Spatial-Temporal Graph Convolutional Network (STGCN) (Yu et al., 2017) is proposed to address the problem of time series traffic prediction by harnessing comprehensive spatial-temporal correlations.
- ASTGNN: The Attention-based Spatial-temporal Graph Neural Network (ASTGNN) (Guo et al., 2021) is designed to capture the dynamics of traffic data across both temporal and spatial dimensions.
- STMFGNN: Spatial-Temporal Multifactor Fusion Graph Neural Network (STMFGNN) (Jia et al., 2024) leverages dynamic similarity and static adjacency graphs for parallel graph convolution, integrating global hidden and local prior knowledge. A gated fusion module adaptively learns dynamic influence weights to capture multiscale spatial dependencies. The model employs gated tanh

unit convolution, multireceptive fields, and gated recurrent units for temporal feature extraction, enabling comprehensive traffic flow prediction by considering multiscale factors.

- OFA: One Fits All (OFA) (Zhou et al., 2023) demonstrates a unified framework for time series prediction by leveraging pretrained language models, specifically GPT-2, without modifying its core architectural components such as self-attention mechanisms or feed-forward residual blocks. We adopt a complementary perspective on traffic data within the OFA framework, optimizing its representation and processing pipeline to enhance predictive performance.
- ST-LLM: Spatial-Temporal Large Language Model (ST-LLM) (Liu et al., 2024) incorporates spatial-temporal embeddings and a fusion convolution layer for unified representation, along with a partially frozen attention mechanism to adapt pretrained LLMs for traffic prediction.
- Llama 2: Llama 2 is a suite of pretrained and fine-tuned large language models developed by Meta, designed for a wide range of natural language understanding and generation tasks. In our baseline setting, we adopt a variant of ST-LLM (Liu et al., 2024) in which the original GPT-2 backbone is replaced with Llama 2 7B.

5.4. Experimental setting

All experiments were conducted using Google Colab, a cloud-based Python environment equipped with an NVIDIA L4 GPU (CUDA 12.4, 23 GB memory) and an Intel® Core™i9-9900KS CPU running at 4.0 GHz. Deep learning models were implemented using the PyTorch framework. For training the LLM-based models, we employed the Ranger21 optimizer, while the Adam optimizer was used for all other deep learning baselines. Hyperparameters, including batch size and learning rate, were carefully tuned through a validation set, with batch sizes ranging from 64 to 256 and learning rates varying between 0.001 and 0.0001, d_{model} is 64. Specifically, the proposed BiSTLLM model utilized a GPT-2 architecture with five transformer layers and a learning rate of 0.001, while the other LLM-based baselines were configured with six GPT-2 layers using the same learning rate. Llama 2 7B is used in the baseline. We implemented the baselines as described in their original paper.

5.5. Full-sample prediction

To assess the effectiveness of the proposed BiSTLLM framework in standard traffic flow prediction tasks, we conduct comprehensive experiments using the full-sample bicycle traffic flow prediction based on the Rotterdam and Leiden datasets. The training time for each epoch of the proposal model is around 3-4 s and inference time is around 0.3-0.4 s. The comparative results against baseline models are presented in Table 2 with average 15 min, 30 min and 60 min prediction errors. Overall, the proposed BiSTLLM exhibits the highest accuracy.

In particular, conventional spatial-temporal deep learning models such as STMFGNN, ASTGCN, and STGCN show comparatively lower performance compared to BiSTLLM. This may be attributed to their limited ability to capture the complex and highly variable spatial-temporal patterns of bicycle traffic. Furthermore, simpler models like HA and LSTM, which either lack spatial modeling or struggle with non-linear temporal dependencies, also exhibit notably lower predictive accuracy.

Among the LLM-based baselines, BiSTLLM outperforms Llama 2, ST-LLM, and OFA, achieving reductions in average MAE of 5.83%, 1.74%, and 17.96% respectively, on the Rotterdam dataset for the 60 min prediction horizon. Similarly, on the Leiden dataset, BiSTLLM achieves MAE reductions of 4.89%, 3.59%, and 23.86%, respectively. These improvements highlight BiSTLLM's enhanced capacity to model complex bicycle traffic dynamics by accurately capturing representative traffic embeddings and effectively fine-tuning the pre-trained LLM. The difference in prediction errors across datasets such as BiSTLLM has a WAPE of 35.70% in Rotterdam compared to 21.09% in Leiden is primarily due to differences in the magnitude and distribution of bicycle traffic volumes between the two cities.

In summary, BiSTLLM demonstrates strong generalization across locations and time scales, offering significant improvements over both traditional deep learning models and existing LLM-based baselines in the bicycle traffic flow prediction task.

5.6. Ablation study

The results of different ablation studies show the impact of including different components in the BiSTLLM modeling framework, as well as the impact of the proposed fine-tuning strategy.

5.6.1. BiSTLLM components ablation study

To assess the contribution of each component within the proposed BiSTLLM framework, we conduct a series of ablation experiments. These experiments systematically include or exclude specific modules to evaluate their individual impact on the model's overall performance in bicycle traffic flow prediction. Specifically, we examine several variants of BiSTLLM, including: (1) w/o Temporal: a variant with the temporal dependency modeling module removed; (2) w/o Spatial: a variant with the spatial dependency modeling module excluded; (3) w/o LLM: a version without the pre-trained LLM component; (4) w/o Lag: a variant omitting the weather lag processing module; (5) w/o Weather: a version in which weather embeddings are removed; (6) w/o POI: a variant without the POI embedding; and (7) w/o Accessibility: a version excluding the bicycle lane accessibility embedding.

Fig. 4 presents the results of the ablation study, highlighting the impact of removing individual components from the BiSTLLM framework. The removal of the pre-trained LLM component (w/o LLM) results in a substantial increase in prediction errors across MAE, RMSE, and WAPE metrics. This suggests that the LLM is a critical element in capturing complex dependencies inherent in bicycle

Table 2
Performance comparison of multi-step traffic prediction (The unit is the count of bicycle traffic per 5 min). The bold results are the best.

Datasets	Baselines	15 min			30 min			60 min		
		MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE
Rotterdam	HA	58.77	81.47	85.95%	58.76	81.46	85.97%	58.76	81.45	86.01%
	LSTM	25.83	42.57	40.75%	27.32	44.95	43.12%	30.24	50.42	47.74%
	STGCN	24.94	39.70	39.35%	25.40	40.58	40.08%	26.59	42.91	41.99%
	ASTGCN	24.20	38.14	38.18%	24.65	38.94	38.90%	26.00	41.63	41.05%
	STMFGNN	23.24	37.04	36.67%	23.59	37.66	37.22%	24.54	39.16	38.73%
	OFA	26.64	43.18	42.04%	26.98	43.83	42.57%	27.56	45.08	43.51%
	ST-LLM	22.48	35.83	35.47%	22.68	36.24	35.79%	23.01	37.13	36.33%
	Llama 2	23.55	36.84	37.16%	23.75	37.22	37.47%	24.01	37.86	37.90%
	BiSTLLM	**22.26**	**35.31**	**35.12%**	**22.34**	**35.47**	**35.26%**	**22.61**	**36.07**	**35.70%**
Leiden	HA	82.26	123.25	73.42%	82.27	123.25	73.42%	82.27	123.26	73.42%
	LSTM	29.23	51.07	26.69%	31.86	56.05	28.76%	36.51	65.61	32.97%
	STGCN	27.13	44.48	24.78%	29.66	50.00	26.78%	33.10	56.96	29.89%
	ASTGCN	24.78	40.84	22.63%	26.60	44.58	24.02%	29.27	50.09	26.43%
	STMFGNN	23.55	38.64	21.51%	25.15	41.50	22.71%	26.71	45.73	24.11%
	OFA	28.02	46.36	25.59%	29.13	49.37	26.30%	30.68	53.09	27.70%
	ST-LLM	22.68	38.42	20.71%	23.38	40.07	21.11%	24.23	43.04	21.88%
	Llama 2	22.69	37.27	20.72%	23.46	39.08	21.19%	24.56	42.20	22.18%
	BiSTLLM	**21.59**	**35.89**	**19.72%**	**22.40**	**37.72**	**20.22%**	**23.36**	**41.18**	**21.09%**

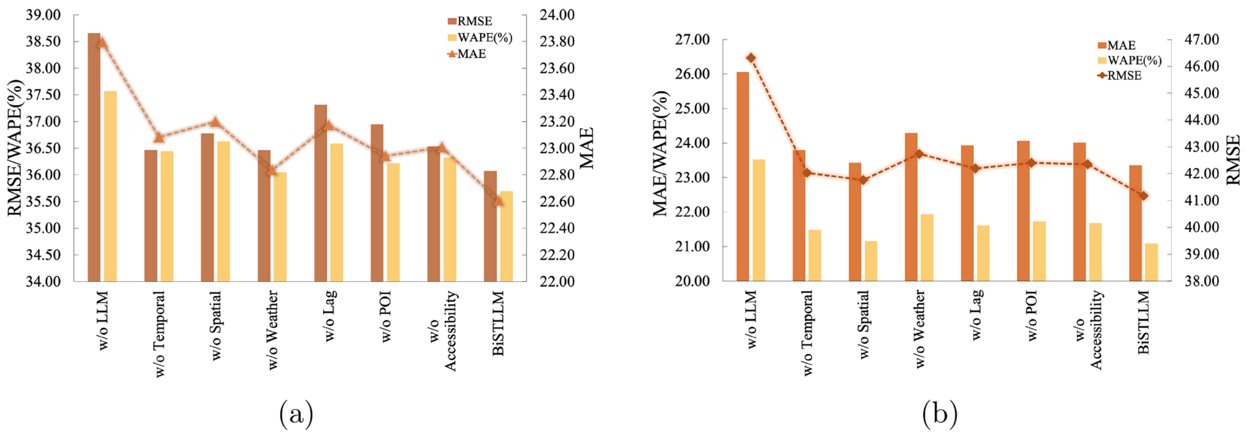

Fig. 4. BiSTLLM components ablation study. (a) Rotterdam. (b) Leiden. Each component contributes positively to the overall performance of the model.

traffic flow data. Furthermore, the exclusion of the temporal (w/o Temporal) and spatial (w/o Spatial) modules also leads to significant performance degradation, underscoring the importance of spatial-temporal representations for accurately modeling traffic dynamics. In addition, removing external contextual embeddings, specifically weather embeddings (w/o Weather), POI embedding (w/o POI), and bicycle lane accessibility embedding (w/o Accessibility), consistently worsens prediction performance. This demonstrates the utility of integrating external factors that influence bicycle traffic patterns. Notably, the w/o Lag variant, which omits the weather lagging mechanism, exhibits a marked increase in error. This may be attributed to a misalignment between weather conditions and their delayed impact on traffic flow; without accounting for this temporal offset, the weather input may act as noise rather than informative context.

5.6.2. Pre-trained LLM fine-tuning strategy ablation study

In this section, we conduct an ablation study to evaluate the effectiveness of our fine-tuning strategy for the pre-trained LLM to do bicycle traffic flow prediction. Our approach builds upon the Partially Frozen Attention (PFA) mechanism proposed by Liu et al. (2024), with key distinctions in whether the positional encoding module and components of the final U layers are frozen or unfrozen.

To systematically assess the impact of these variations, we designed a set of model configurations by selectively freezing or unfreezing different components. The experimental variants are as follows: (1) Frozen-Full: A fully frozen model where no layers or modules are fine-tuned. (2) Frozen-Pe & Frozen-Att-Proj(ours): The positional encoding module is frozen. Additionally, within the final U layers, the query, key, and value projection matrices of the attention mechanism are kept frozen. (3) Unfrozen-Pe & Frozen-Att-Proj: The positional encoding module is unfrozen, while the attention projections in the last U layers remain frozen. (4) Unfrozen-Pe & Unfrozen-Att-Proj: Both the positional encoding module and the attention projection matrices in the last U layers are

Table 3
Ablation study of fine-tuning strategy (average 60 min Horizon).

Datasets	Baselines	MAE	RMSE	WAPE
Rotterdam	Unfrozen-Pe & Frozen-Att-Proj	22.86	36.69	36.08%
	Unfrozen-Pe & Unfrozen-Att-Proj	22.81	36.25	36.00%
	Frozen-Full	22.79	36.31	35.97%
	Frozen-Pe & Unfrozen-Att-Proj	22.77	36.53	35.94%
	Frozen-Pe & Frozen-Att-Proj (Ours)	**22.61**	**36.07**	**35.70%**
Leiden	Unfrozen-Pe & Frozen-Att-Proj	23.54	41.74	21.26%
	Unfrozen-Pe & Unfrozen-Att-Proj	23.83	42.18	21.51%
	Frozen-Full	23.42	41.34	21.15%
	Frozen-Pe & Unfrozen-Att-Proj	23.83	42.53	21.52%
	Frozen-Pe & Frozen-Att-Proj (Ours)	**23.36**	**41.18**	**21.09%**

Table 4
Performance comparison of few-shot traffic prediction. The bold results are the best.

Datasets	Baselines	15 min			30 min			60 min		
		MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE
Rotterdam	LSTM	33.92	57.94	53.52%	34.85	58.46	54.99%	37.21	62.09	58.74%
	STGCN	37.35	60.73	58.93%	38.51	61.41	60.77%	43.88	67.69	69.28%
	ASTGCN	32.94	55.09	51.97%	36.52	56.76	57.63%	41.13	61.50	64.94%
	STMFGNN	26.20	42.79	41.34%	26.92	43.86	42.49%	28.77	47.23	45.42%
	OFA	28.88	46.67	45.56%	30.01	48.94	47.35%	32.09	53.55	50.66%
	ST-LLM	25.75	41.89	40.63%	25.97	42.02	40.98%	27.09	44.04	42.77%
	Llama 2	29.13	45.00	45.96%	30.02	46.53	47.37%	31.88	49.68	50.33%
	BiSTLLM	**25.23**	**41.79**	**39.80%**	**25.37**	**41.76**	**40.04%**	**26.24**	**43.28**	**41.43%**
Leiden	LSTM	84.40	138.31	77.07%	85.74	140.23	77.42%	85.83	140.49	77.50%
	STGCN	82.69	131.77	75.51%	84.62	135.45	76.41%	86.68	139.52	78.27%
	ASTGCN	66.10	105.35	60.36%	73.79	109.52	66.63%	80.91	114.10	73.05%
	STMFGNN	27.04	48.16	24.70%	29.22	51.72	26.38%	32.19	57.05	29.06%
	OFA	30.32	51.31	27.69%	32.40	55.59	29.26%	35.85	63.72	32.37%
	ST-LLM	29.22	49.87	26.68%	29.71	50.94	26.82%	30.48	52.32	27.52%
	Llama 2	30.55	45.92	27.90%	30.59	47.76	27.62%	31.98	51.00	28.88%
	BiSTLLM	**24.89**	**41.59**	**22.73%**	**25.83**	**43.50**	**23.32%**	**27.37**	**47.21**	**24.72%**

unfrozen, allowing maximum adaptation. (5) Frozen-Pe & Unfrozen-Att-Proj: The positional encoding module remains frozen, while the attention projection matrices in the final U layers are fine-tuned.

The ablation results, presented in Table 3, demonstrate that freezing both the positional encoding and the attention projection layers yields the best performance among all fine-tuning strategies. In contrast, variants that involve unfreezing either the positional encoding or the attention projection layers result in increased prediction errors. One possible explanation is that the pre-trained LLM in our framework primarily leverages spatial attention mechanisms, which are not inherently dependent on any sequential order among nodes. Since traffic nodes lack a natural ordering, unfreezing the positional encoding may introduce noise or confusion into the model's spatial reasoning. Moreover, while the pre-trained projections were originally optimized for linguistic tasks, their generic ability to model pairwise interactions (via query/key/value mappings) may transfer to traffic nodes. Freezing these layers preserves this inductive bias, preventing overfitting to limited fine-tuning data.

5.7. Parameter analysis

The performance of the BiSTLLM framework is notably sensitive to its hyperparameter settings. To identify the optimal configuration for bicycle traffic flow prediction, we conducted a comprehensive grid search across multiple hyperparameter combinations. As illustrated in Fig. 5, we investigated various configurations of the pre-trained LLM in terms of total number of layers and the number of unfrozen layers U used during fine-tuning. Motivated by previous findings (Zhou et al., 2023), which highlight a favorable trade-off between performance and computational cost with a 6-layer GPT-2 variant, we examined models with 5, 6, and 7 layers. We further evaluated the impact of varying the number of unfrozen layers U on predictive performance. The results, presented in Fig. 5c and f, reveal that the best performance was achieved using a 5-layer LLM with 4 unfrozen layers on the Rotterdam dataset, and a 5-layer LLM with 5 unfrozen layers on the Leiden dataset. These findings suggest that deeper fine-tuning of shallower LLMs can effectively enhance domain adaptation for traffic prediction tasks. Additionally, we performed a grid search on the batch size and learning rate. As shown in Fig. 5a, d, b, and e, we observed that a batch size of 64 and a learning rate of 0.001 consistently yielded the highest prediction accuracy on both datasets. These hyperparameter settings were used in all subsequent experiments.

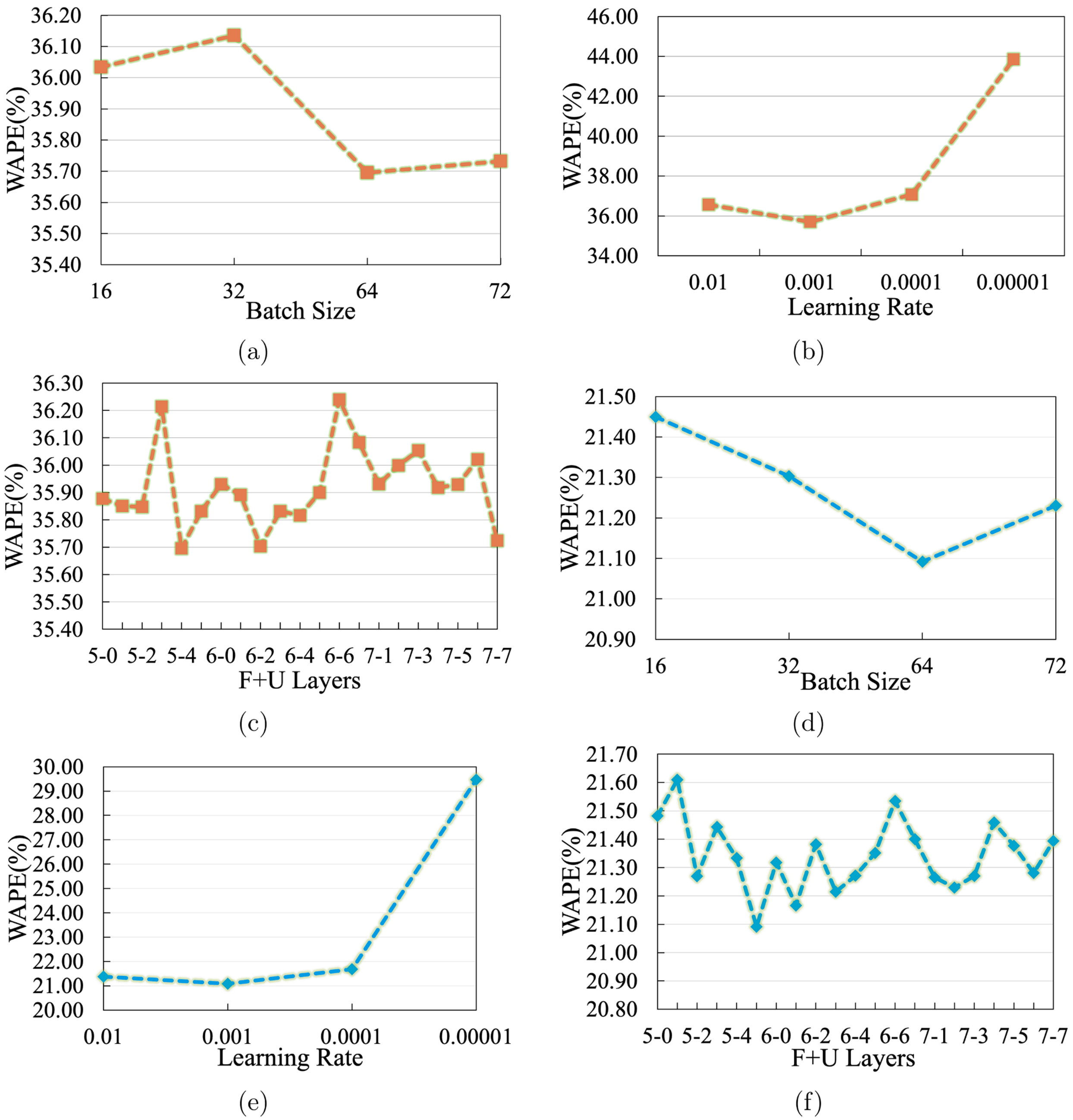

Fig. 5. Parameter analysis. (a) Batch Size(Rotterdam). (b) Learning Rate(Rotterdam). (c) $F + U$ layers(Rotterdam). (d) Batch Size(Leiden). (e) Learning Rate(Leiden). (f) $F + U$ layers(Leiden). Here $F + U$ is the total number of layers and U is the number of unfrozen layers by allowing updates to the attention computations in attention mechanisms and to the LN sublayers during fine-tuning.

5.8. *Few-shot prediction*

Collecting high-quality bicycle traffic data presents significant challenges due to both technical and resource limitations. The prediction task is defined as using the past one hour of observations to predict the next one hour of traffic flow. Unlike car traffic, which often benefits from extensive sensing infrastructure and continuous monitoring systems, bicycle traffic monitoring typically relies on fewer and often less accurate sensors deployed in limited locations. Often, these sensors operate only during specific time periods to reduce operational costs. As a result, the data collected is often sparse, inconsistent, or incomplete over time. In such scenarios, traditional data-hungry prediction models may underperform or fail to generalize. To address this limitation, we adopt a few-shot learning framework, which enables effective traffic flow prediction from limited training samples, thereby improving model adaptability in real-world, low-resource settings.

Consistent with the full-sample experiments, the datasets are chronologically divided into training, validation, and testing sets. However, for the few-shot prediction setting, only last 10% of the original training set is used, while the validation and test sets remain identical to those used in the full-sample experiments. Consequently, the final data split corresponds to 6% for training, 20% for validation, and 20% for testing. As reported in Table 4, the average prediction errors across 15 min, 30 min, and 60 min horizons demonstrate that BiSTLLM consistantly outperforms other baselines, specifically, BiSTLLM outperforms Llama 2, ST-LLM, and OFA, achieving reductions in average MAE of 17.68%, 3.13%, and 18.22%, respectively, on the Rotterdam dataset for the 60 min prediction horizon. Similarly, on the Leiden dataset, BiSTLLM achieves MAE reductions of 14.40%, 10.17%, and 23.63%, respectively. These results underscore the robustness ability of BiSTLLM, particularly in data-scarce scenarios.

6. Conclusions

Bicycle traffic exhibits significant fluctuations due to its strong sensitivity to external environmental factors. This inherent variability makes it considerably less predictable than vehicular traffic. To address these challenges and effectively capture the fluctuations in bicycle traffic flow, we have proposed BiSTLLM, a novel framework for bicycle traffic flow prediction based on pre-trained LLM.

In the BiSTLLM framework, we introduce a weather-lagging processor and incorporate semantic embeddings of POIs and bicycle lane accessibility, enabling the model to learn infrastructure-dependent traffic behaviors and localized activity patterns. These enhancements facilitate the model's ability to capture both local and global spatial-temporal interactions. To adapt the general knowledge encoded in a pre-trained LLM to the domain of bicycle traffic, we implement a domain-specific fine-tuning strategy. Experiments demonstrate that BiSTLLM consistently outperforms existing state-of-the-art models across various prediction horizons. Given that bicycle traffic data is often sparse, seasonal, and unevenly distributed over time, we also evaluate BiSTLLM under few-shot prediction scenarios, where only 10% of the training data is used. Notably, BiSTLLM maintains strong performance in these low-data conditions, highlighting its robustness and adaptability.

In conclusion, this study illustrates the potential of leveraging pre-trained LLM for accurate bicycle traffic prediction, even in data-sparse environments. However, several limitations remain. One key omission is the assumption that limited historical data can be supplemented effectively through fine-tuning based on pre-trained LLM without fully evaluating the differences of urban contexts. Additionally, due to constraints in data availability and scope, we were unable to evaluate the model's performance across a wider network of cities. While BiSTLLM is designed as a general spatiotemporal framework and does not rely on region-specific features, the experimental validation in this study is limited to bicycle traffic datasets from the Netherlands. Differences in cycling behavior, infrastructure, differences of temporal segments, and data collection practices across regions may affect model performance.

For future work, exploring the computational complexity and time cost of the proposed model in real world practice, performing extensive multi-seed experiments for statistical significance testing and further developing a generalized bicycle traffic prediction model capable of predicting flow at sensor locations with no historical data are necessary. Additionally, examining the interpretability of the proposed model with directional neworks through transportation theory is necessary to better understand its functioning. In addition, including the evaluation to more diverse cycling datasets from other cities, countries, or different temporal stratification is an important direction to assess the generalizability and scalability of the proposed approach. Addressing these challenges will further enhance the real-world applicability of the bicycle traffic prediction model, contributing to the broader goal of providing effective solutions in data-scarce scenarios and supporting more sustainable transportation planning.

Statement

During the preparation of this work the corresponding author used ChatGPT in the writing process to improve the readability and language of the manuscript. After using this tool, the corresponding author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

CRediT authorship contribution statement

Xiamei Wen: Writing – review & editing, Writing – original draft, Visualization, Validation, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization; **Dorine Duives:** Writing – review & editing, Supervision, Software, Resources, Project administration, Methodology, Investigation; **Serge Hoogendoorn:** Writing – review & editing, Supervision, Software, Resources, Project administration, Investigation, Funding acquisition, Data curation, Conceptualization.

Declaration of competing interest

None.

Acknowledgement

This research is supported by the China Scholarship Council. This support is gratefully acknowledged.

Data availability

Data will be made available on request.

References

Awad, M., Khanna, R., Awad, M., Khanna, R., 2015. Support vector regression. Effic. Learn. Mach. Theor. Concepts Appl. Eng. Syst. Des. 67–80.
Bai, S., Kolter, J.Z., Koltun, V., 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271.
Böcker, L., Dijst, M., Prillwitz, J., 2013. Impact of everyday weather on individual daily travel behaviours in perspective: a literature review. Transp. Rev. 33 (1), 71–91.
Box, G. E.P., Pierce, D.A., 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. J. Am. Stat. Assoc. 65 (332), 1509–1526.
Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32.
Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078
Dill, J., Carr, T., 2003. Bicycle commuting and facilities in major US cities: if you build them, commuters will use them. Transp. Res. Rec. 1828 (1), 116–123.
Graves, A., Graves, A., 2012. Long short-term memory. Supervised Seq. Label. Recurr. Neural Netw. 385, 37–45.
Guo, S., Lin, Y., Wan, H., Li, X., Cong, G., 2021. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. IEEE Trans. Knowl. Data Eng. 34 (11), 5415–5428.
Hanson, S., Hanson, P., 1977. Evaluating the impact of weather on bicycle use. Transp. Res. Rec. 629, 43–48.
Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
Huang, X., 2024. Enhancing traffic prediction with textual data using large language models. arXiv preprint arXiv:2405.06719.
Izanloo, A., Rafsanjani, A.K., Ebrahimi, S.P., 2017. Effect of commercial land use and accessibility factor on traffic flow in bojnourd. J. Urban Plan. Dev. 143 (2), 05016016.
Janiesch, C., Zschech, P., Heinrich, K., 2021. Machine learning and deep learning. Electron. Mark. 31 (3), 685–695.
Ji, A., Li, D., Dai, Z., Cui, M., Yu, L., Duan, Z., 2025a. A hybrid graph memory network approach with multi-level feature representation for traffic flow forecast. Expert Syst. Appl. 300, 130316.
Ji, A., Liu, Z., Su, L., Dai, Z., 2025b. A hybrid framework for spatio-temporal traffic flow prediction with multi-scale feature extraction. Inf. Sci. 716, 122259.
Jia, H., Yu, Z., Chen, Y., Xia, H., 2024. Spatial-temporal multi-factor fusion graph neural network for traffic prediction. Appl. Intell. 54 (19), 9464–9487.
Jiang, Y., Pan, Z., Zhang, X., Garg, S., Schneider, A., Nevmyvaka, Y., Song, D., 2024. Empowering time series analysis with large language models: a survey. arXiv preprint arXiv:2402.03182.
Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J.Y., Shi, X., Chen, P.-Y., Liang, Y., Li, Y.-F., Pan, S., et al., 2023. Time-LLM: time series forecasting by reprogramming large language models. arXiv preprint arXiv:2310.01728.
Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.
Lee, X., Gu, Y., 2024. Step-LLM: spatial-temporal-enriched prompting with large language models for traffic prediction. In: International Conference of Electrical, Electronic and Networked Energy Systems. Springer, pp. 248–255.
Li, J., Liu, C., Cheng, S., Arcucci, R., Hong, S., 2024a. Frozen language model helps ecg zero-shot learning. In: Medical Imaging with Deep Learning. PMLR, pp. 402–415.
Li, P., Pang, Y., Ren, J., 2024b. Spatio-temporal graph convolutional network combined large language model: a deep learning framework for bike demand forecasting. arXiv preprint arXiv:2403.15733.
Li, Z., Cao, Y., Xu, X., Jiang, J., Liu, X., Teo, Y.S., Lin, S.-W., Liu, Y., 2024c. Llms for relational reasoning: how far are we? In: Proceedings of the 1st International Workshop on Large Language Models for Code, pp. 119–126.
Liu, C., Yang, S., Xu, Q., Li, Z., Long, C., Li, Z., Zhao, R., 2024. Spatial-temporal large language model for traffic prediction. In: 2024 25th IEEE International Conference on Mobile Data Management (MDM). IEEE, pp. 31–40.
Liu, J., Ong, G.P., Chen, X., 2020. GraphSAGE-based traffic speed forecasting for segment network with sparse data. IEEE Trans. Intell. Transp. Syst. 23 (3), 1755–1766.
Lu, K., Grover, A., Abbeel, P., Mordatch, I., 2022. Frozen pretrained transformers as universal computation engines. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36, pp. 7628–7636.
Naghib, A., Jafari Navimipour, N., Hosseinzadeh, M., Sharifi, A., 2023. A comprehensive and systematic literature review on the big data management techniques in the internet of things. Wirel. Netw. 29 (3), 1085–1144.
Ojekudo, N.A., Akpan, N.P., 2017. Anapplication of Dijkstra's algorithm to shortest route problem. IOSR J. Math. 13 (3), 14.
O'shea, K., Nash, R., 2015. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.
Qi, X., Yao, J., Wang, P., Shi, T., Zhang, Y., Zhao, X., 2024. Combining weather factors to predict traffic flow: a spatial-temporal fusion graph convolutional network-based deep learning approach. IET Intell. Transp. Syst. 18 (3), 528–539.
Qu, H., Li, H., You, L., Zhu, R., Yan, J., Santi, P., Ratti, C., Yuen, C., 2024. ChatEV: predicting electric vehicle charging demand as natural language processing. Transp. Res. D Transp. Environ. 136, 104470.
Ramezani, A., Xu, Y., 2023. Knowledge of cultural moral norms in large language models. arXiv preprint arXiv:2306.01857.
Ren, Y., Chen, Y., Liu, S., Wang, B., Yu, H., Cui, Z., 2024. TPLLM: a traffic prediction framework based on pretrained large language models. arXiv preprint arXiv:2403.02221.
Shi, X., Qi, H., Shen, Y., Wu, G., Yin, B., 2020. A spatial–temporal attention approach for traffic prediction. IEEE Trans. Intell. Transp. Syst. 22 (8), 4909–4918.
Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P., 2013. The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process. Mag. 30 (3), 83–98.
Song, C.H., Wu, J., Washington, C., Sadler, B.M., Chao, W.-L., Su, Y., 2023. LLM-planner: few-shot grounded planning for embodied agents with large language models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2998–3009.
Ton, D., Duives, D.C., Cats, O., Hoogendoorn-Lanser, S., Hoogendoorn, S.P., 2019. Cycling or walking? Determinants of mode choice in the Netherlands. Transp. Res. Part A Policy Pract. 123, 7–23.
Torrey, L., Shavlik, J., 2010. Transfer learning. In: Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques. IGI Global, pp. 242–264.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30, 5998–6008.
VM, K., Warrier, H., Gupta, Y., et al., 2024. Fine tuning LLM for enterprise: practical guidelines and recommendations. arXiv preprint arXiv:2404.10779.
Wen, X., Khosla, M., Hoogendoorn, S., 2025a. Multi-source transfer learning with spatial-temporal graph neural network for short-term bicycle traffic prediction. IEEE Trans. Intell. Transp. Syst. 27, 1952–1968.
Wen, X., Krishnakumari, P., Hoogendoorn, S., 2025b. Dynamic spatial-temporal graph convolutional neural networks approach for active mode traffic prediction. IEEE Trans. Intell. Transp. Syst. 26, 16496–16510.
Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S., 2020. A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. 32 (1), 4–24.
Xu, H., Sharaf, A., Chen, Y., Tan, W., Shen, L., Van Durme, B., Murray, K., Kim, Y.J., 2024. Contrastive preference optimization: pushing the boundaries of LLM performance in machine translation. arXiv preprint arXiv:2401.08417.

Xu, Y., Liu, M., 2025. Gpt4tfp: spatio-temporal fusion large language model for traffic flow prediction. Neurocomputing 625, 129562.

Yao, H., Tang, X., Wei, H., Zheng, G., Li, Z., 2019. Revisiting spatial-temporal similarity: a deep learning framework for traffic prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33, pp. 5668–5675.

Yao, X., Calvert, S.C., Hoogendoorn, S.P., 2025. Driving heterogeneity identification using machine learning: a review and framework for analysis. Transp. Res. Interdiscip. Perspect. 32, 101511.

Ye, J., Xue, S., Jiang, A., 2022. Attention-based spatio-temporal graph convolutional network considering external factors for multi-step traffic flow prediction. Digit. Commun. Netw. 8 (3), 343–350.

Ye, J., Zhang, W., Yi, K., Yu, Y., Li, Z., Li, J., Tsung, F., 2024. A survey of time series foundation models: generalizing time series representation with large language model. arXiv preprint arXiv:2405.02358.

Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., Yin, B., 2021. Deep learning on traffic prediction: methods, analysis, and future directions. IEEE Trans. Intell. Transp. Syst. 23 (6), 4927–4943.

Yu, B., Yin, H., Zhu, Z., 2017. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. Research Square preprint, DOI: 10.21203/rs.3.rs-5380374/v1.

Yu, X., Chen, Z., Ling, Y., Dong, S., Liu, Z., Lu, Y., 2023. Temporal data meets LLM–explainable financial time series forecasting. arXiv preprint arXiv:2306.11025.

Zhao, R., Sun, Z., Ji, A., 2022. A deep reinforcement learning approach for automated on-ramp merging. In: 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 3800–3806.

Zhou, T., Niu, P., Sun, L., Jin, R., et al., 2023. One fits all: power general time series analysis by pretrained LM. Adv. Neural Inf. Process. Syst. 36, 43322–43355.

Zollner, D., Vasiliev, R., Castellano, B., Molnar, G., Janssen, P., 2024. A technical examination to explore conditional multimodal contextual synthesis in large language models. Research Square preprint. https://doi.org/10.21203/rs.3.rs-5380374/v1.