GAT-LLM: A Graph Attention-Based Framework with Large Language Models for Spatio-Temporal Traffic Forecasting

Sadia Tabassum[a], Sumon Ahmed[a] and Naushin Nower[a;*]

[a]Institute of Information Technology, University of Dhaka
ORCiD ID: Sadia Tabassum https://orcid.org/0009-0005-4745-2762,
Sumon Ahmed https://orcid.org/0000-0002-4618-5018, Naushin Nower https://orcid.org/0000-0001-5640-829X

Abstract. Traffic forecasting is a crucial component of intelligent transportation systems (ITS), with applications in congestion prevention, route optimization, and urban mobility management. Recent advances in traffic forecasting have focused on combining graph-based spatial modeling and large language model (LLM) based temporal reasoning. In this work, we introduce **GAT-LLM**, a novel hybrid architecture that integrates Graph Attention Networks (GATs) for spatial representation with a partially frozen pre-trained LLM for temporal sequence modeling. GATs effectively capture spatial dependencies across traffic locations, while LLM handles long-range temporal patterns in traffic sequences. We propose a unified embedding strategy that fuses graph-derived spatial features, temporal encodings, and positional embeddings, creating semantically rich and well-structured input for transformer-based models. Our approach constructs a spatio-temporal representation of traffic conditions, enabling context-aware input for pretrained LLMs. Preliminary experiments on benchmark datasets show promising results, outperforming existing graph-based, attention-based, and LLM-based models. This work highlights the potential of combining graph neural networks and LLMs to build more expressive and transferable spatiotemporal forecasting systems. The code is available at: https://github.com/SadiaTabassum1216/GATLLM.

1 Introduction

Traffic congestion is a major global issue with a detrimental impact on economic efficiency and quality of life. In certain countries, the costs associated with it are estimated to be around 2% of GDP [2]. Traffic forecasting predicts congestion levels based on various methods to optimize flow and support informed route design. Accurate forecasting is the foundation of intelligent transportation systems (ITS), with applications in congestion prevention, route optimization, and urban mobility management. It helps users plan routes, saves time and energy, and enables city planners to manage traffic effectively. The core challenge of traffic prediction lies in capturing the complex spatiotemporal nature of traffic data. This includes dynamic temporal variations such as rush hour patterns and daily/weekly cycles, and spatial correlations governed by the road network topology. Additionally, traffic flow is influenced by social events, holidays, weather conditions, road accidents, and other external factors [20].

Traditional statistical models such as ARIMA [1] and Kalman filters [12] capture temporal patterns but fail to model the complex, non-linear, and non-stationary nature of traffic flow with spatiotemporal dependencies. Deep learning techniques address this issue, as they can model such complex and dynamic characteristics. Earlier DNN-based approaches use convolutional neural networks (CNNs) for spatial and recurrent neural networks (RNNs) for temporal dependencies. However, CNNs are designed for grid-shaped structures and use Euclidean distance, making them less suitable for non-Euclidean road networks [10].

To address the problem of CNN, graph neural networks (GNNs) are effective models for learning spatial dependencies, representing roads as nodes and their connections as edges. When combined with temporal models such as LSTM, GRU, and TCNs, frameworks like DCRNN [14] and STGCN [28] effectively model spatiotemporal dependencies. However, in many cases, the spatial and temporal components are either fused shallowly or treated separately, limiting generalization in complex urban settings. Graph Convolutional Networks (GCNs) [11] use fixed weights for neighborhood aggregation, while Graph Attention Networks (GATs) [24] use dynamic attention mechanisms based on node relevance, making them better at capturing heterogeneous traffic patterns.

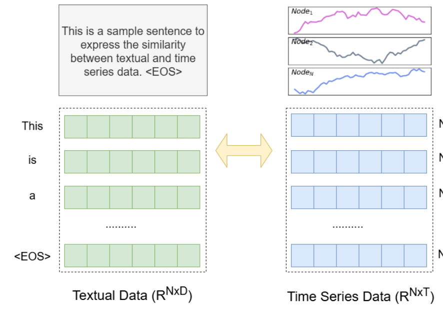
This is a sample sentence to express the similarity between textual and time series data. <EOS>
Node$_1$
Node$_2$
Node$_N$
This
is
a
....
<EOS>
Node$_1$
Node$_2$
Node$_3$
....
Node$_N$
Textual Data (R^{NxD})
Time Series Data (R^{NxT})

Figure 1. Similarity between textual data and time series data

Recently, large language models (LLMs) have achieved significant success across domains such as NLP, computer vision, and time series analysis. Figure 1 illustrates the structural similarity between time series and textual data. Although LLMs were originally

[*] Corresponding Author. Email: naushin@iit.du.ac.bd.

designed for NLP using token embeddings, not directly suited for transportation time-series inputs, their ability to process sequences of fixed dimensional vectors makes them suitable for multivariate traffic data [21]. This opens new opportunities for traffic forecasting. We focus on LLMs due to their strong generalization and reasoning abilities, enabling versatile application with minimal task-specific training. While their effectiveness in other domains is well established, few studies have explored their use for modeling urban dynamics from spatiotemporal data.

Architectures like Informer [31], Autoformer [25], and FEDformer [32] leverage attention mechanisms and frequency-aware designs to enhance time series forecasting. LLMs such as GPT-2 and BERT have been adapted to time-series forecasting by tokenizing numeric sequences. Models like LLMTime [6] and PromptCast [27] exemplify this approach, showing strong temporal modeling capabilities including few-shot and zero-shot generalization. However, they mainly focus on temporal aspects and lack spatial topology modeling. STLLM [17] processes multi-location sequences but does not incorporate graph-based spatial dependencies, limiting its ability to understand traffic flows with road connectivity. Some LLMs like GATGPT [5] integrate graph neural networks for spatial dependency. While GATGPT combines graph attention for spatial relationships with LLM-based temporal modeling using positional embeddings, its design primarily emphasizes spatial integration, with less attention to tailored long-range temporal modeling.

To address these limitations, we introduce GAT-LLM, a novel architecture that combines Graph Attention Networks (GATs) for spatial representation with a pretrained language model for temporal sequence modeling. Our approach constructs a unified spatiotemporal representation by embedding road relationships through GATs and fusing them with learnable temporal encodings and positional embeddings. This representation is fed into a partially frozen transformer: lower layers remain frozen to retain general sequence modeling capabilities, while higher layers are fine-tuned to adapt to traffic specific patterns, balancing generalization and specialization. The key contributions of our work are as follows:

- We propose **GAT-LLM**, a hybrid spatio-temporal model that integrates graph-based spatial learning with large language model-based temporal reasoning for traffic forecasting.
- We design a unified embedding strategy that fuses graph attention-based spatial features, temporal encodings, and positional embeddings, allowing structured and context-aware input to pre-trained LLMs.
- We use a partially frozen fine-tuning mechanism for the LLM, enabling efficient training. This approach leverages general purpose reasoning from the lower transformer layers while adapting the higher layers to the target domain.

This work bridges the gap between spatial graph modeling and transformer-based sequence learning in traffic forecasting, offering a new paradigm for developing scalable, accurate, and transferable spatiotemporal models. The remainder of this paper is organized as follows. Section 2 reviews related work, Section 3 presents the proposed GAT-LLM architecture, Section 4 describes the experimental setup and results, and Section 5 concludes the paper.

2 Related Work

In this section, we review related work on traffic prediction from two perspectives: graph-based architectures and language model-based architectures.

Graph-Based Architectures: Recent advancements in spatio-temporal traffic forecasting have focused on graph-based methods, which are effective in modeling road network spatial topology. Graph Convolutional Networks (GCNs) have gained popularity for capturing spatial dependencies. For example, DCRNN [14] introduced diffusion convolution for spatio-temporal modeling, while STGCN [28] combined spectral graph convolutions with 1D convolution for better feature extraction. Other notable GCN-based models include T-GCN [29], which integrates GCNs with Gated Recurrent Units (GRUs) to capture both spatial and temporal characteristics, and A3TGCN [3], which enhances T-GCN by adding attention to detect global traffic flow trends. ASTGCN [7] employs an attention mechanism to weigh the importance of different time steps and spatial neighbors. PSTGCN[10] further addresses these limitations of static adjacency matrices by employing a dynamic probabilistic spatiotemporal graph. However, GCNs aggregate information using fixed weights, limiting their ability to capture dynamic spatial relationships and long-range temporal dependencies, particularly under dynamic conditions such as road closures or accidents. To address these issues, Graph Attention Networks (GATs) introduce dynamic attention to better model spatial dependencies in dynamic traffic networks. GMAN [30] extends this by employing adaptive attention and gating to capture both spatial and temporal dynamics, while STGAT [8] combines GAT with an additional LSTM to jointly learn spatial interactions and their temporal correlations. This highlights the need for models that integrate both spatial and temporal dependencies for effective traffic forecasting. Despite these advancements, effectively integrating both complex spatial and long-range temporal dependencies remains a challenge in many graph-based approaches, highlighting a critical area for further research.

Language Model-Based Architectures: The success of Transformer architectures[23] in natural language processing has led to their adaptation for time series forecasting. Transformer-based models like Informer [31], Autoformer [25], and FEDformer [32] have achieved state-of-the-art results in time series forecasting by improving attention efficiency and capturing periodicity. Building upon this, Large language models (LLMs), such as GPT-2 and BERT, have recently been adapted for numeric time series forecasting by tokenizing data, enabling pre-trained models to perform zero-shot or fine-tuned forecasting tasks. This is evident in models like LLMTime [6], Time-LLM [9], and PromptCast [27]. GATGPT [5], for example, leverages the generative capabilities of LLMs while integrating graph attention for spatial dependencies, although focusing primarily on spatial aspects over comprehensive temporal modeling.

UrbanGPT [15] and UniTime [18] extend LLMs to cross-domain forecasting through instruction tuning and in-context learning. TFT [16] is another model that combines LSTM with attention mechanisms to capture temporal dependencies, making it effective in multi-horizon forecasting. However, models like UrbanMind [19], which utilize prompt engineering, often fail to capture crucial spatial dependencies, which are essential in dynamic traffic networks.

STLLM [17] introduces a partially frozen transformer for multi-location forecasting, yet lacks explicit graph-based spatial modeling, limiting its ability to capture key spatial dynamics such as road connectivity and neighborhood interactions. While these models excel in temporal generalization, they remain spatially agnostic, which restricts their overall effectiveness in traffic forecasting. This gap underscores the need for novel architectures that can effectively combine the powerful temporal modeling capabilities of LLMs with robust spatial representations to fully capture the complexities of spatio-temporal traffic data.

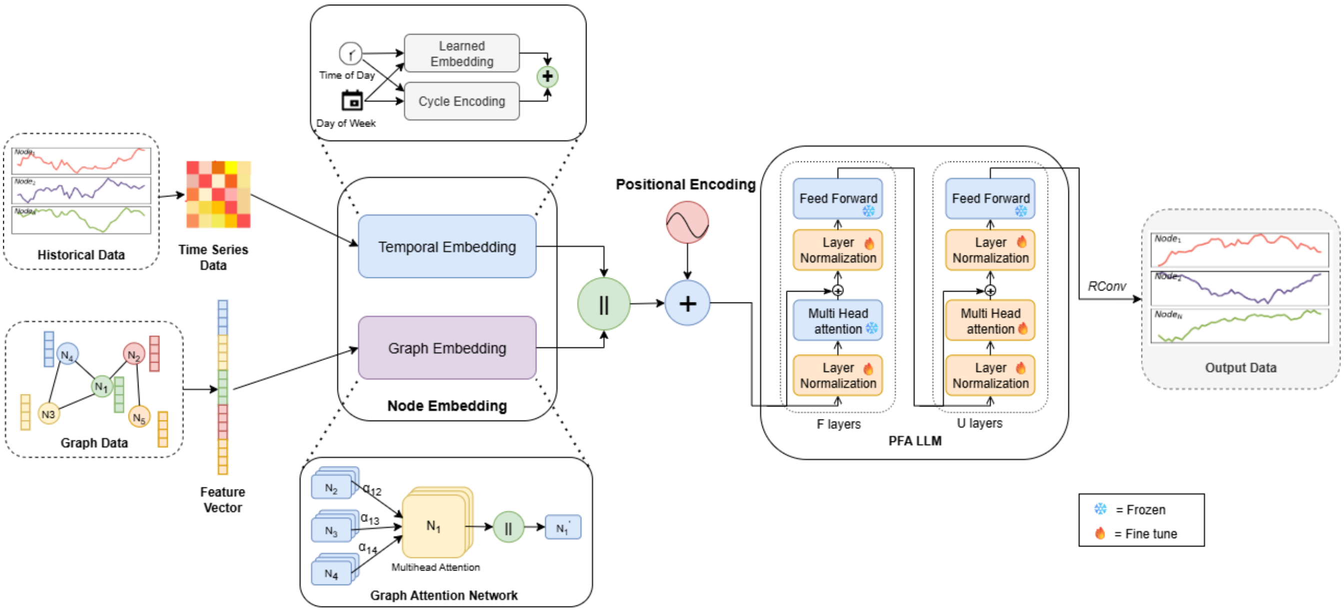

Figure 2. Architecture of our proposed GAT-LLM model

3 Methodology

In this section, the problem statement and detailed structure of the proposed GAT-LLM are described.

3.1 *Problem Statement*

The traffic prediction problem can be defined as a spatio-temporal forecasting problem. The road network is modeled as a graph $G = (V, E, A)$, where: V is the set of N road segments, E is the set of edges representing the connectivity of the road, $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix encoding the connections between nodes. At each time step t, traffic measurements across all road segments are captured by a feature vector $\mathbf{x}_t \in \mathbb{R}^N$, where the n-th element represents the measurement from road segment n. Given historical data over the past T time steps, $X = \{\mathbf{x}_{t-T+1}, \dots, \mathbf{x}_t\} \in \mathbb{R}^{N\times T}$, along with the adjacency matrix A. The goal of the traffic prediction task is to predict the traffic data for the next T' time steps. The model outputs the predicted traffic values for the next T' time steps as $Y = f(X, A)$, where $Y = \{\mathbf{y}_{t+1}, \dots, \mathbf{y}_{t+T'}\} \in \mathbb{R}^{N\times T'}$ represents the predicted sequence of traffic features over those future time steps. In this study, we approximate $f(\cdot)$ using our GAT-LLM model, which combines temporal embedding, graph embedding, and a partially frozen pretrained LLM for the prediction task.

3.2 *Details of proposed GAT-LLM*

The proposed GAT-LLM is a fusion-based architecture for spatio-temporal traffic forecasting that integrates temporal embeddings, graph-based spatial reasoning, and a partially frozen pretrained language model. As shown in Figure 2, the model captures both temporal patterns and spatial dependencies through a structured processing pipeline.

Temporal embeddings are constructed using a combination of learnable intra/inter-day components and sinusoidal encoding from the historical data to capture daily and weekly trends. In parallel, spatial relationships among traffic nodes are modeled using a multi-head Graph Attention Network (GAT), which enables each node to focus on its most relevant neighbors via learned attention weights. The resulting temporal and spatial embeddings are concatenated to form node-level representations.

These node embeddings are then added with learnable positional encoding to retain temporal sequence and passed into a Large Language Model (LLM). Here, the lower transformer layers remain frozen, while the upper attention layers are fine-tuned to adapt to traffic-specific dynamics. This partially frozen attention strategy balances generalization with task-specific learning, enabling efficient and accurate traffic prediction.

3.3 *Temporal Embedding*

To effectively model periodic patterns in time series data, we incorporate both learnable and sinusoidal temporal embeddings. The time series historical data serves as the input for constructing these embeddings. The learnable embeddings consist of two components: one capturing *intra-day* variations across different time steps within a day, and another capturing *inter-day* patterns across the seven days of the week.

In parallel, we construct *cycle-based embeddings* using sine and cosine transformations to represent temporal periodicity:

$$\mathrm{Cycle}_{\mathrm{day},t} = \sin\left(\frac{2\pi t}{T_d}\right) + \cos\left(\frac{2\pi t}{T_d}\right) \quad (1)$$

$$\mathrm{Cycle}_{\mathrm{week},t} = \sin\left(\frac{2\pi t}{T_w}\right) + \cos\left(\frac{2\pi t}{T_w}\right) \quad (2)$$

where, T_d denotes the number of time steps in a day and T_w denotes the number of days in a week. These scalar values are then projected into the embedding space through learnable linear transformations.

The final temporal representation is obtained by summing the learned intraday and interday embeddings with the projected cycle-based components:

$$\mathbf{e}_i^{(t)} = \mathbf{e}_{\text{intra-day},t} + \mathbf{e}_{\text{inter-day},t} + \mathbf{e}_{\text{cycle-day},t} + \mathbf{e}_{\text{cycle-week},t} \quad (3)$$

This design enables the model to effectively capture both fixed temporal patterns and periodic trends across different time scales.

3.4 Graph Embedding

To better understand how traffic patterns are influenced by nearby locations, we generate graph-based embeddings that encode the structural properties of the road network. These embeddings enable the model to capture how traffic on one road segment is affected by its neighbors. We employ a multi-head graph attention network (GAT) to learn these spatial relationships. GAT assigns dynamic attention weights to each neighboring node, allowing the model to focus more on the most influential connections.

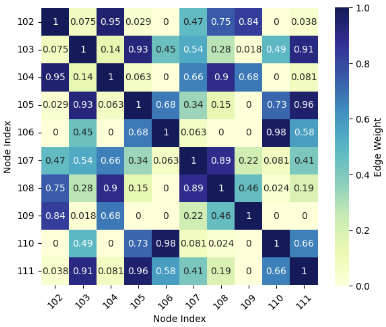

Figure 3. Spatial correlation among ten randomly selected neighboring nodes

Figure 3 presents an adjacency heatmap showing the learned spatial dependencies among ten selected neighboring nodes from the NYC taxi dataset. The heatmap reveals significant variation in influence strength, highlighting that real-world spatial dependencies are often heterogeneous and context-dependent.

Let $\mathbf{x}_i \in \mathbb{R}^C$ denote the input feature vector of node i, where C is the feature dimension. In GAT, the attention coefficient between node i and its neighbor j is computed as:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top[\mathbf{W}\mathbf{x}_i \parallel \mathbf{W}\mathbf{x}_j]\right)\right)}{\sum_{k\in\mathcal{N}(i)} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top[\mathbf{W}\mathbf{x}_i \parallel \mathbf{W}\mathbf{x}_k]\right)\right)} \tag{4}$$

Here, $\mathbf{x}_i$ denotes the input feature vector of node i, $\mathbf{W} \in \mathbb{R}^{F\times C}$ is a shared learnable weight matrix for feature projection, $\mathbf{a} \in \mathbb{R}^{2F}$ is a learnable attention vector, α_{ij} is the attention weight from node i to neighbor j, and $\mathcal{N}(i)$ is the set of neighbors of node i. $\|$ denotes concatenation, and $\mathbf{a}^\top$ is the transpose of the attention vector. To enhance learning capacity and stability, we apply multi-head attention by computing K attention heads in parallel and concatenating their outputs:

$$\mathbf{e}_i^{(s)} = \|_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(k)} \mathbf{W}^{(k)} \mathbf{x}_j\right) \tag{5}$$

Here, $\alpha_{ij}^{(k)}$ and $\mathbf{W}^{(k)}$ denote the attention weights and projection matrices for the k-th head, σ is a nonlinear activation function, and $\|$ indicates concatenation. The resulting vector $\mathbf{e}_i^{(s)}$ is the spatial embedding for node i, capturing its neighborhood structure.

In our model, the GAT module is applied once at the beginning to generate fixed spatial embeddings for each node, based on the road network's adjacency matrix. These embeddings are shared across all time steps, enabling consistent spatial context. We combine them with the input features and temporal encoding to form spatiotemporal representations, which are then processed by the LLM module to model long-range temporal dependencies. This decoupled spatial-temporal learning framework improves efficiency while preserving the model's ability to capture complex interactions across space and time.

3.5 Positional Encoding

To preserve the temporal structure of the input sequence, we introduce a learnable positional encoding. Since the model bypasses the default node and positional embeddings of traditional transformer-based architectures, explicit encoding is required to maintain sequence order. This positional encoding is designed to match the dimensionality of the pre-trained LLM, ensuring compatibility during processing.

The input road data $\mathbf{X} \in \mathbb{R}^{T\times N\times C}$ consists of T time steps, N road segments, and feature dimension C. The temporal embedding of node i is denoted $\mathbf{e}_i^{(t)}$, and the spatial embedding is $\mathbf{e}_i^{(s)}$. These are concatenated to form a combined representation:

$$\mathbf{e}_{t,i}^{(st)} = \left[\mathbf{e}_i^{(t)} \| \mathbf{e}_i^{(s)}\right] \tag{6}$$

A learnable positional encoding $\mathbf{P}_{t,i}$ is then added to the combined embeddings at each time step t and node i:

$$\mathbf{X'}_{t,i} = \mathbf{e}_{t,i}^{(st)} + \mathbf{P}_{t,i}, \quad \forall t \in \{1, T\},\ i \in \{1, N\} \tag{7}$$

This addition enables the model to distinguish between different time steps and retain the sequential nature of the input.

3.6 Partially Frozen Attention (PFA) LLM

We adopt a *Partially Frozen Attention* (PFA) strategy to integrate pretrained knowledge from GPT-2 while adapting to the spatial-temporal characteristics of traffic data. Unlike the Frozen Pretrained Transformer (FPT), which freezes all transformer layers, PFA selectively unfreezes only the attention modules in the upper layers. This allows task-specific fine-tuning with minimal modification, retaining the generalization ability of the pretrained model.

The model takes as input a combined representation that encodes spatial, temporal, and positional information for each road segment. We denote the full encoded sequence $\mathbf{X} = \{\mathbf{X'}_{t,i}\}$ as the initial hidden representation H_0 for the input to the LLM. This representation is fed into a truncated version of the LLM with L layers, where the first $F = L - U$ layers are fully frozen—including both multi-head attention (MHA) and feed-forward network (FFN) modules. The remaining U layers retain frozen FFNs but unfreeze MHA modules to allow adaptation. All layers follow the standard Transformer structure with pre-layer normalization and residual connections:

$$\bar{H}_i = \text{MHA}(\text{LN}(H_i)) + H_i, \tag{8}$$

$$H_{i+1} = \text{FFN}(\text{LN}(\bar{H}_i)) + \bar{H}_i, \quad i \in [1, L] \tag{9}$$

Here, the trainable or frozen status of the MHA and FFN modules depends on the layer index i. For $i \leq F$, both MHA and FFN are frozen; for $i > F$, only MHA is trainable. The detailed module definitions are:

$$\text{LN}(H_i) = \gamma \odot \frac{H_i - \mu}{\sigma} + \beta, \tag{10}$$

$$\text{MHA}(\tilde{H}_i) = W^O(\text{head}_1 \| \cdots \| \text{head}_h), \tag{11}$$

$$\begin{aligned}\text{head}_j &= \text{Attn}(W_j^Q \tilde{H}_i, W_j^K \tilde{H}_i, W_j^V \tilde{H}_i), && (12)\\ \text{Attn}(Q,K,V) &= \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, && (13)\\ \text{FFN}(\hat{H}_i) &= \Big(\max(0, \hat{H}_i W_1 + b_1)\Big) W_2 + b_2 && (14)\end{aligned}$$

Here, $\tilde{H}_i$ and $\hat{H}_i$ denote the outputs of the first and second layer normalization, respectively; γ and β are learnable parameters of LN; μ and σ denote the batch-wise mean and standard deviation; $\odot$ indicates element-wise multiplication, and $\|$ denotes concatenation across attention heads.

To stabilize training and improve information flow, a residual connection is added between the input and output of the LLM. Finally, a regression head is applied via a convolutional layer to predict traffic features over the next T' time steps:

$$\hat{Y}_{T'} = \text{RConv}(H_{\text{final}}; \theta_r), \quad \hat{Y}_{T'} \in \mathbb{R}^{T' \times N \times C} \tag{15}$$

Here, H_{final} represents the final LLM output for all time steps, θ_r denotes the parameters of the regression head, N is the number of road segments, and C is the number of output channels. This design enables efficient fine-tuning on domain-specific spatial-temporal patterns while preserving the expressive power of the pretrained model.

4 Experiments

In this section, our objective is to validate the superiority of our GAT-LLM through a series of extensive experimental evaluations.

4.1 Datasets

This section details the datasets used to examine the predictive performance of GAT-LLM and baselines, with real-world traffic data from NYCTaxi and CHBike.

NYCTaxi: The NYCTaxi dataset captures over 35 million taxi rides across New York City, neatly organized into 266 virtual stations. Covering the period from April 1st to June 30th, 2016, it records data in half-hour intervals, resulting in a total of 4,368 timesteps.

CHBike: The CHBike dataset tracks how Citi Bike bike-sharing program in New York City used the bike-sharing system during the same three-month period as NYCTaxi, from April to June 2016. After filtering out less-used stations, it focuses on the 250 busiest ones. Like NYCTaxi, it breaks down the data into 4,368 half-hour intervals.

4.2 Baselines

We compare *GAT-LLM* against a diverse set of strong baselines, classified into two main groups based on their architectural design-Graph-based models and LLM-based models.

Graph-based Models: These models focus on learning spatial and temporal dependencies directly from graph-structured data without using LLM.

- **DCRNN** [14]: Models traffic flow as a diffusion process over a graph, employing diffusion convolution within a recurrent framework.
- **STGCN** [28]: Combines spectral graph convolutions with temporal 1D convolutions to jointly model spatial and temporal dependencies.
- **ASTGCN** [7]: Integrates spatial and temporal attention mechanisms with graph convolution for adaptive modeling of traffic patterns.
- **GWN** [26]: Introduces an adaptive adjacency matrix and gated temporal convolutions for flexible spatial modeling.
- **AGCRN** [4]: Employs adaptive graph learning and node-specific parameters to model heterogeneous spatio-temporal dependencies.
- **GMAN** [30]: An attention-based model with spatial and temporal gating units in an encoder-decoder structure.
- **ASTGNN** [7]: Employs hierarchical attention to capture both node-level and time-level importance.
- **DGCRN** [13]: A dynamic graph convolutional recurrent network that learns evolving graph structures over time.

LLM-based Models: These models leverage large language models (LLMs) for time series forecasting tasks, adapting them to the spatio-temporal context.

- **OFA** [33]: A unified transformer based model, adapted for all types of time series tasks, avoids making changes to the self-attention and feed-forward layers inside GPT-2's residual blocks.
- **LLAMA2** [22]: A large pretrained transformer-based language model developed by Meta. We adapt LLAMA2 to the traffic forecasting setting using frozen transformer layers and time-series input formatting.
- **GATGPT** [5]: Combines the Graph Attention Network (GAT) with a frozen pretrained GPT-2 transformer to model spatiotemporal dependencies for imputation task.
- **STLLM** [17]: A recent LLM-based approach that encodes multi-location time series as token sequences and leverages a partially frozen transformer for forecasting.

4.3 Implementations

Aligning with contemporary practices, we split the NYCTaxi and CHBike datasets into training, validation, and test sets using a 6:2:2 ratio. The historical input length P and prediction horizon S were both set to 12, enabling multi-step traffic forecasting. Weekly periodicity was modeled using $T_w = 7$, and daily periodicity using $T_d = 48$, where each step corresponds to 30 minutes.

Experiments for *GAT-LLM* were conducted on Kaggle's GPU platform, which offered limited computational capacity. The underlying language model was GPT-2 with six transformer layers, trained autoregressively—i.e., it predicts one time step at a time, feeding the output of the previous step into the next. For comparison, we followed the baseline setup reported in the STLLM paper [17]. All models used the Adam optimizer with a learning rate of 0.001, as reported by the original authors. Our evaluation protocols were aligned with those in STLLM to ensure a fair comparison.

4.4 Evaluation Metrics

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

Table 1. Model comparison in terms of MAE, RMSE, MAPE (%), and WAPE (%).

Models	**NYCTaxi Pick-up**				**NYCTaxi Drop-off**				**CHBike Pick-up**				**CHBike Drop-off**			
	MAE	RMSE	MAPE	WAPE	MAE	RMSE	MAPE	WAPE	MAE	RMSE	MAPE	WAPE	MAE	RMSE	MAPE	WAPE
DCRNN	5.40	9.71	35.09%	20.43%	5.19	9.63	37.78%	19.82%	2.09	3.30	54.22%	42.26%	1.96	2.94	51.42%	39.61%
STGCN	5.71	10.22	36.51%	21.62%	5.38	9.60	39.12%	20.55%	2.08	3.31	53.63%	42.08%	2.01	3.07	50.45%	40.62%
ASTGCN	7.43	13.84	47.96%	28.04%	6.98	14.70	45.48%	26.60%	2.76	4.45	64.23%	55.71%	2.79	4.20	69.88%	56.49%
GWN	5.43	$\underline{9.39}$	37.79%	20.55%	5.03	8.78	35.63%	19.21%	2.04	3.20	53.08%	40.95%	1.95	2.98	50.30%	39.43%
AGCRN	5.79	10.11	40.40%	21.93%	5.45	9.56	40.67%	20.81%	2.16	3.46	56.35%	43.69%	2.06	3.19	51.91%	41.78%
GMAN	5.43	9.47	34.39%	20.42%	5.09	8.95	35.00%	19.33%	2.20	3.35	57.34%	44.06%	2.09	3.00	54.82%	42.00%
ASTGNN	5.90	10.71	40.15%	22.32%	6.28	12.00	49.78%	23.97%	2.37	3.67	60.08%	47.81%	2.24	3.35	57.21%	45.27%
DGCRN	5.44	9.82	35.78%	20.58%	5.14	$\underline{9.39}$	35.09%	19.64%	2.06	3.21	54.06%	41.51%	1.96	2.93	51.99%	39.70%
OFA	5.82	10.42	36.67%	22.00%	5.60	10.14	37.39%	21.36%	2.06	3.21	53.55%	41.70%	1.96	2.97	49.64%	39.68%
GATGPT	5.92	10.55	37.83%	22.39%	5.66	10.39	37.36%	21.60%	2.07	3.23	**52.54%**	41.70%	1.95	2.94	**49.26%**	39.43%
LLAMA2	5.35	9.48	41.32%	20.27%	5.66	10.74	47.47%	21.63%	2.10	3.37	56.63%	42.49%	1.99	3.03	55.23%	40.28%
ST-LLM	$\underline{5.29}$	9.42	**33.55%**	$\underline{20.03\%}$	**5.07**	**9.07**	**33.34%**	$\underline{19.18\%}$	$\underline{1.99}$	$\underline{3.08}$	53.54%	**40.19%**	$\underline{1.89}$	$\underline{2.81}$	$\underline{49.50\%}$	$\underline{38.27\%}$
GAT-LLM	**5.24**	**9.21**	$\underline{34.08\%}$	**18.27%**	$\underline{5.12}$	9.43	$\underline{34.25\%}$	**18.03%**	**1.58**	**2.45**	$\underline{52.78\%}$	$\underline{40.29\%}$	**1.51**	**2.25**	50.12%	**28.13%**

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

$$MAPE = \frac{100}{m}\sum_{i=1}^{m}\left|\frac{\hat{Y}_i - Y_i}{Y_i}\right|, \tag{17}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\hat{Y}_i - Y_i\right)^2}, \tag{18}$$

$$WAPE = \frac{\sum_{i=1}^{m}\left|\hat{Y}_i - Y_i\right|}{\sum_{i=1}^{m}\left|Y_i\right|} \times 100. \tag{19}$$

where m is the number of all predicted values.

4.5 Parameter Analysis

To enhance *GAT-LLM*'s performance, we applied Bayesian Optimization to tune key hyperparameters. The final configuration included a learning rate of 1×10^{-4}, batch size of 8, and 300 training epochs. A weight decay of 3×10^{-3} was used to improve generalization. All experiments were conducted on Kaggle's hosted environment, equipped with two NVIDIA Tesla T4 GPUs (each with 16 GB VRAM), 31 GB of system RAM, and CUDA version 12.6. Due to RAM constraints, a smaller batch size was necessary for stable training. Additionally, only the top 2 transformer layers were fine-tuned, while lower layers remained frozen to balance computational efficiency with task-specific adaptation.

4.6 Results

The comparison results with baselines are shown in Table 1. The bold results are the best, and the underlined results are the second best. Result on taxi drop dataset can be seen in Figure 4.

The experimental evaluation across multiple benchmark datasets reveals several key insights into spatio-temporal traffic forecasting:

- The proposed **GAT-LLM** achieves state-of-the-art performance across all scenarios, consistently outperforming LLM-based baselines such as **OFA** and **LLAMA2**, highlighting its superior ability to capture complex spatio-temporal patterns.
- Compared to **STLLM**, which lacks explicit graph modeling, **GAT-LLM** shows clear improvements across nearly all metrics, confirming the benefit of integrating graph attention for spatial dependencies.

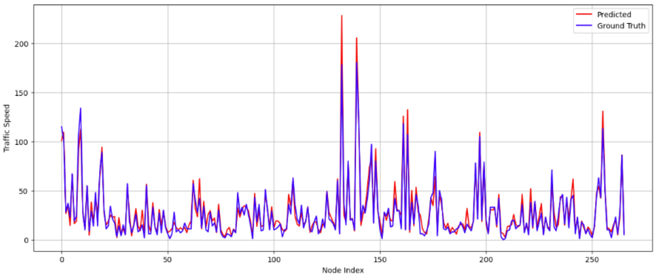

Figure 4. Traffic speed prediction at 30-minute horizon compared with ground truth in Taxi_Drop dataset

- Larger model size does not guarantee better performance. Despite having more parameters, **LLAMA2** underperforms **GAT-LLM**, emphasizing the importance of task-specific design over scale.
- **OFA**, while a strong generalist model, performs poorly on structured traffic data due to its limited handling of spatio-temporal dependencies.
- Graph-based models like **GWN** and **DGCRN** capture spatial patterns well but struggle with temporal generalization, resulting in lower overall performance than **GAT-LLM**.
- Attention-based GNNs such as **ASTGNN** and **GMAN** perform well on specific datasets but lack the consistency of **GAT-LLM**, indicating limitations in generalization.

Overall, the results suggest a clear performance hierarchy among model categories, with LLM-based architectures outperforming graph-based counterparts. The consistent superiority of the proposed **GAT-LLM** highlights the effectiveness of integrating graph-based spatial reasoning with LLM-driven temporal modeling, offering a robust and generalizable solution for spatio-temporal traffic forecasting.

5 Conclusion

In this work, we introduced **GAT-LLM**, a novel hybrid architecture that integrates graph-based spatial modeling with large language model-based temporal reasoning for spatio-temporal traffic forecasting. By combining Graph Attention Networks (GATs) with a partially frozen pretrained transformer, GAT-LLM is designed to jointly capture spatial dependencies across traffic locations and long-range temporal patterns in traffic sequences. Our unified embedding strategy incorporates graph-derived spatial features, temporal encodings, and positional embeddings, making the input well-structured and semantically meaningful for transformer-based processing.

Preliminary experiments on benchmark datasets such as *NYCTaxi* and *CHBike* suggest that GAT-LLM offers promising performance compared to existing graph-based, attention-based, and LLM-based models, including STLLM, OFA, and LLAMA2. While these initial results are encouraging, further evaluation is needed to comprehensively assess generalization and scalability across diverse traffic environments. Our study highlights the potential of aligning GNNs with LLMs to build more expressive and transferable spatio-temporal forecasting systems. Future work may explore broader benchmarks and applications in real-time intelligent transportation systems, where domain-aware structures can enhance the capabilities of pretrained sequence models.

References

[1] Taghreed Alghamdi, Khalid Elgazzar, Magdi Bayoumi, Taysseer Sharaf, and Sumit Shah, 'Forecasting traffic congestion using arima modeling', in *2019 15th international wireless communications & mobile computing conference (IWCMC)*, pp. 1227–1232. IEEE, (2019).

[2] Mehdi Attioui and Mohamed Lahby, 'Congestion forecasting using machine learning techniques: A systematic review', *Future Transportation*, **5**(3), 76, (2025).

[3] Jiandong Bai, Jiawei Zhu, Yujiao Song, Ling Zhao, Zhixiang Hou, Ronghua Du, and Haifeng Li, 'A3t-gcn: Attention temporal graph convolutional network for traffic forecasting', *ISPRS International Journal of Geo-Information*, **10**(7), 485, (2021).

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

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

[6] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson, 'Large language models are zero-shot time series forecasters', *Advances in Neural Information Processing Systems*, **36**, 19622–19635, (2023).

[7] Shengnan Guo, Youfang Lin, Huaiyu Wan, Xiucheng Li, and Gao Cong, 'Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting', *IEEE Transactions on Knowledge and Data Engineering*, **34**(11), 5415–5428, (2021).

[8] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang, 'Stgat: Modeling spatial-temporal interactions for human trajectory prediction', in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6272–6281, (2019).

[9] 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., 'Time-llm: Time series forecasting by reprogramming large language models', *arXiv preprint arXiv:2310.01728*, (2023).

[10] Atkia Akila Karim and Naushin Nower, 'Probabilistic spatio-temporal graph convolutional network for traffic forecasting', *Applied Intelligence*, **54**(11), 7070–7085, (2024).

[11] TN Kipf, 'Semi-supervised classification with graph convolutional networks', *arXiv preprint arXiv:1609.02907*, (2016).

[12] Selvaraj Vasantha Kumar, 'Traffic flow prediction using kalman filtering technique', *Procedia Engineering*, **187**, 582–587, (2017).

[13] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li, 'Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution', *ACM Transactions on Knowledge Discovery from Data*, **17**(1), 1–21, (2023).

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

[15] Zhonghang Li, Lianghao Xia, Jiabin Tang, Yong Xu, Lei Shi, Long Xia, Dawei Yin, and Chao Huang, 'Urbangpt: Spatio-temporal large language models', in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5351–5362, (2024).

[16] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister, 'Temporal fusion transformers for interpretable multi-horizon time series forecasting', *International journal of forecasting*, **37**(4), 1748–1764, (2021).

[17] Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao, 'Spatial-temporal large language model for traffic prediction', in *2024 25th IEEE International Conference on Mobile Data Management (MDM)*, pp. 31–40. IEEE, (2024).

[18] Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann, 'Unitime: A language-empowered unified model for cross-domain time series forecasting', in *Proceedings of the ACM Web Conference 2024*, pp. 4095–4106, (2024).

[19] Yuhang Liu, Yingxue Zhang, Xin Zhang, Ling Tian, Yanhua Li, and Jun Luo, 'Urbanmind: Urban dynamics prediction with multifaceted spatial-temporal large language models', in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1951–1962, (2025).

[20] Md Moshiur Rahman and Naushin Nower, 'Attention based deep hybrid networks for traffic flow prediction using google maps data', in *Proceedings of the 2023 8th international conference on machine learning technologies*, pp. 74–81, (2023).

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

[22] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al., 'Llama 2: Open foundation and fine-tuned chat models', *arXiv preprint arXiv:2307.09288*, (2023).

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, 'Attention is all you need', *Advances in neural information processing systems*, **30**, (2017).

[24] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio, 'Graph attention networks', *arXiv preprint arXiv:1710.10903*, (2017).

[25] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long, 'Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting', *Advances in neural information processing systems*, **34**, 22419–22430, (2021).

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

[27] Hao Xue and Flora D Salim, 'Promptcast: A new prompt-based learning paradigm for time series forecasting', *IEEE Transactions on Knowledge and Data Engineering*, **36**(11), 6851–6864, (2023).

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

[29] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li, 'T-gcn: A temporal graph convolutional network for traffic prediction', *IEEE transactions on intelligent transportation systems*, **21**(9), 3848–3858, (2019).

[30] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi, 'Gman: A graph multi-attention network for traffic prediction', in *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 1234–1241, (2020).

[31] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang, 'Informer: Beyond efficient transformer for long sequence time-series forecasting', in *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, (2021).

[32] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin, 'Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting', in *International conference on machine learning*, pp. 27268–27286. PMLR, (2022).

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