



UrbanGPT: Spatio-Temporal Large Language Models

Zhonghang Li
South China University of Technology
The University of Hong Kong
Guangzhou & Hong Kong SAR, China
bjdwh.zzh@gmail.com

Lianghao Xia
The University of Hong Kong
Hong Kong SAR, China
aka_xia@foxmail.com

Jiabin Tang
The University of Hong Kong
Hong Kong SAR, China
jiabintang77@gmail.com

Yong Xu
South China University of Technology
Guangzhou, China
yxu@scut.edu.cn

Lei Shi
Baidu Inc.
Beijing, China
harryshi.cs@gmail.com

Long Xia
Baidu Inc.
Beijing, China
long.phil.xia@gmail.com

Dawei Yin
Baidu Inc.
Beijing, China
yindawei@acm.org

Chao Huang*
The University of Hong Kong
Hong Kong SAR, China
chaohuang75@gmail.com

Abstract

Spatio-temporal prediction aims to forecast and gain insights into the ever-changing dynamics of urban environments across both time and space. Its purpose is to anticipate future patterns, trends, and events in diverse facets of urban life, including transportation, population movement, and crime rates. Although numerous efforts have been dedicated to developing neural network techniques for accurate predictions on spatio-temporal data, it is important to note that many of these methods heavily depend on having sufficient labeled data to generate precise spatio-temporal representations. Unfortunately, the issue of data scarcity is pervasive in practical urban sensing scenarios. In certain cases, it becomes challenging to collect any labeled data from downstream scenarios, intensifying the problem further. Consequently, it becomes necessary to build a spatio-temporal model that can exhibit strong generalization capabilities across diverse spatio-temporal learning scenarios.

Taking inspiration from the remarkable achievements of large language models (LLMs), our objective is to create a spatio-temporal LLM that can exhibit exceptional generalization capabilities across a wide range of downstream urban tasks. To achieve this objective, we present the UrbanGPT, which seamlessly integrates a spatio-temporal dependency encoder with the instruction-tuning paradigm. This integration enables LLMs to comprehend the complex inter-dependencies across time and space, facilitating more comprehensive and accurate predictions under data scarcity. To validate the effectiveness of our approach, we conduct extensive experiments on

various public datasets, covering different spatio-temporal prediction tasks. The results consistently demonstrate that our UrbanGPT, with its carefully designed architecture, consistently outperforms state-of-the-art baselines. These findings highlight the potential of building large language models for spatio-temporal learning, particularly in zero-shot scenarios where labeled data is scarce. The code and data are available at: <https://github.com/HKUDS/UrbanGPT>.

CCS Concepts

- **Information systems** → **Data mining**: *Spatial-temporal systems*;
- **Computing methodologies** → **Artificial intelligence**.

Keywords

Large Language Models; Spatial-Temporal Data Mining; Urban Computing; Smart Cities; Generative AI

ACM Reference Format:

Zhonghang Li, Lianghao Xia, Jiabin Tang, Yong Xu, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. 2024. UrbanGPT: Spatio-Temporal Large Language Models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*, August 25–29, 2024, Barcelona, Spain. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3637528.3671578>

1 Introduction

Spatio-temporal prediction is driven by the motivation to accurately forecast and gain valuable insights into the dynamic nature of urban environments. By analyzing and understanding the ever-changing dynamics across both time and space, spatio-temporal prediction allows us to anticipate future patterns, trends, and events in various aspects of urban life. This has significant implications in the field of urban computing, where the ability to predict transportation patterns can optimize traffic flow, reduce congestion, and enhance overall urban mobility [19, 32]. Moreover, anticipating population movement can aid in effective urban planning and resource allocation [7, 21]. Additionally, the ability to forecast crimes can greatly contribute to enhancing public safety [33]. Spatio-temporal prediction plays a vital role in shaping smarter and more efficient cities, ultimately leading to an improved quality of urban life.

*Chao Huang is the Corresponding Author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '24, August 25–29, 2024, Barcelona, Spain.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0490-1/24/08

<https://doi.org/10.1145/3637528.3671578>

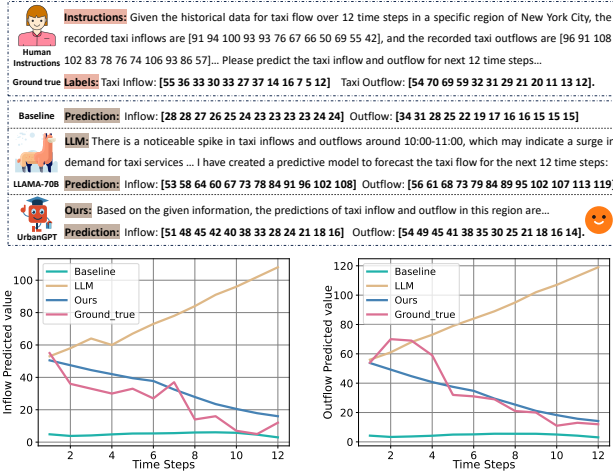


Figure 1: The superior predictive performance of the proposed UrbanGPT compared to the large language model (LLaMA-70B) and the spatiotemporal graph neural network (STGCN) in a zero-shot traffic flow prediction scenario.

It is important to highlight the various types of neural network architectures commonly adopted in this domain of spatio-temporal prediction. These architectures are designed to capture and model the complex relationships between spatial and temporal dimensions in the data. One widely employed architecture is the Convolutional Neural Network (CNN) [15, 41, 47], which is effective in extracting spatial features by applying convolutional filters across the input data. Another line of spatio-temporal neural networks is the Recurrent Neural Network (RNN) family [1, 35, 45]. Those spatio-temporal RNNs are well-suited for capturing temporal dependencies by maintaining a memory state that can retain information over time. Recently, there has been a surge in the use of Graph Neural Networks (GNNs) for spatio-temporal prediction [38, 42, 49]. GNNs excel in modeling complex spatial relationships in data represented as graphs, where each node corresponds to a spatial location and edges capture the connections between them.

While current spatio-temporal neural network techniques have proven to be highly effective, it is crucial to acknowledge their strong dependence on having an abundance of labeled data in order to generate accurate predictions. However, the pervasive problem of data scarcity in practical urban sensing scenarios poses a significant challenge. For example, deploying sensors throughout the entire urban space to monitor citywide traffic volume or air quality is impractical due to the high cost involved [18, 43]. Moreover, the challenge of limited labeled data availability extends to spatio-temporal forecasting across different cities, in which acquiring labeled data for each target city becomes a daunting task [13, 40]. These issues emphasize the pressing need for novel solutions that address data scarcity and enhance the generalization capabilities of spatio-temporal models in various smart city applications.

Inspired by the remarkable progress of large language models (LLMs), our primary goal is to create a spatio-temporal LLM that possesses outstanding generalization capabilities across a wide array of urban tasks. Leveraging the reasoning abilities inherent in LLMs, we aim to expand their success into the domain of spatio-temporal analysis. Our objective is to develop a model that can

effectively comprehend and forecast intricate spatial and temporal patterns, enabling it to excel in various urban scenarios.

While it is of utmost importance to develop a versatile spatio-temporal model capable of effectively handling diverse downstream tasks, aligning the spatio-temporal context with the knowledge space of large language models (LLMs) and enabling them to comprehend the complex dependencies across time and space present significant challenges. These hurdles call for meticulous model design to bridge the gap between the unique characteristics of spatio-temporal data and the knowledge encoded within LLMs.

Contributions. In light of these challenges, we propose UrbanGPT, a large language model specifically tailored for spatio-temporal prediction. At the core of UrbanGPT lies a novel spatio-temporal instruction-tuning paradigm that seeks to align the intricate dependencies of time and space, with the knowledge space of LLMs. Within our UrbanGPT framework, we start by incorporating a spatio-temporal dependency encoder, which utilizes a multi-level temporal convolutional network. This encoder enables the model to capture the intricate temporal dynamics present in the spatio-temporal data across various time resolutions. Then, our model involves aligning textual and spatio-temporal information to empower language models in effectively injecting spatio-temporal contextual signals. This is achieved through the utilization of a lightweight alignment module that projects the representations of spatio-temporal dependencies. The result is the generation of more expressive semantic representations by integrating valuable information from both textual and spatio-temporal domains.

Through the incorporation of spatio-temporal information during the instruction-tuning process, the language model gains proficiency in understanding and processing the intricate relationships and patterns found in spatio-temporal data. By leveraging the insights obtained from the spatio-temporal domain, the language model becomes better equipped to capture the nuances and complexities of spatio-temporal phenomena. This, in turn, enables the model to make more reliable and insightful predictions across various urban scenarios, even when faced with limited data availability.

To showcase the superior predictive performance of our proposed model, we compare it with the large language model (LLaMA-70B) and the spatio-temporal graph neural network (STGCN) in a zero-shot traffic flow prediction scenario guided by textual instructions, as depicted in Figure 1. The large language model, LLaMA, effectively infers traffic patterns from the input text. However, its limitations in handling numeric time-series data with complex spatial and temporal dependencies can sometimes lead to opposite traffic trend predictions. On the other hand, the pre-trained baseline model demonstrates a strong understanding of spatio-temporal dependencies. However, it may suffer from overfitting to the source dataset and underperform in zero-shot scenarios, indicating its limited generalization capabilities beyond existing spatio-temporal prediction models. In contrast, our proposed model achieves a harmonious integration of domain-specific spatio-temporal knowledge and language modeling capabilities. This enables us to make more accurate and reliable predictions under data scarcity.

In summary, our main contributions can be outlined as follows:

- To the best of our knowledge, this is the first attempt to develop a spatio-temporal large language model capable of predicting

diverse urban phenomena across different datasets, especially under conditions of limited data availability.

- We propose UrbanGPT, a spatio-temporal prediction framework that empowers large language models (LLMs) to comprehend the intricate inter-dependencies across time and space. This is achieved through the seamless integration of a spatio-temporal dependency encoder with the instruction-tuning paradigm, effectively aligning the spatio-temporal context with LLMs.
- Extensive experiments conducted on three benchmark datasets provide compelling evidence of our proposed UrbanGPT's exceptional ability to generalize in zero-shot spatio-temporal learning scenarios. These findings highlight the model's robust generalization capacity, demonstrating its effectiveness in accurately predicting and understanding spatio-temporal patterns, even in scenarios where no prior training data is available.

2 Preliminaries

Spatio-Temporal Data. Spatio-temporal data is commonly collected and can be represented as a three-dimensional tensor $\mathbf{X} \in \mathbb{R}^{R \times T \times F}$. Each element $\mathbf{X}_{r,t,f}$ in the tensor corresponds to the value of the f -th feature at the t -th time interval in the r -th region. To provide an example, let's consider predicting taxi traffic patterns in an urban area. In this scenario, the data can represent the inflow and outflow of taxis in a specific region (e.g., the r -th spatial area) during a given time period from t to $t-1$ (e.g., a 30-minute interval).

Spatio-Temporal Forecasting. In spatio-temporal prediction tasks, a common scenario involves forecasting future trends using historical data. Specifically, the goal is to predict the data for the next P time steps based on information from the preceding H steps.

$$\mathbf{X}_{t_{K+1}:t_{K+P}} = f(\mathbf{X}_{t_K-H+1:t_K}) \quad (1)$$

The function $f(\cdot)$ represents a spatio-temporal prediction model that has been trained effectively using historical data. Spatio-temporal prediction tasks can be divided into two main categories: regression prediction, which involves predicting continuous values like traffic flow or taxi demand [23], and classification prediction, where the goal is to classify events such as crime occurrence prediction [11]. To optimize the model $f(\cdot)$, different loss functions are utilized based on the specific characteristics of the spatio-temporal scenarios.

Spatio-Temporal Zero-Shot Learning. Despite the effectiveness of current spatio-temporal learning approaches, they often encounter difficulties in effectively generalizing across a wide range of downstream spatio-temporal learning scenarios. In this study, our focus is on addressing the challenge of spatio-temporal zero-shot scenarios, where we aim to learn from previously unseen data in downstream spatio-temporal prediction datasets or tasks. This can be formally defined as follows:

$$\tilde{\mathbf{X}}_{t_{K+1}:t_{K+P}} = \hat{f}(\tilde{\mathbf{X}}_{t_K-H+1:t_K}) \quad (2)$$

In this particular scenario, the prediction function $\hat{f}(\cdot)$ is responsible for forecasting the spatio-temporal data $\tilde{\mathbf{X}}$ from downstream tasks that have not been previously encountered. It should be noted that the model $\hat{f}(\cdot)$ is not trained specifically on the target data.

3 Methodology

3.1 Spatio-Temporal Dependency Encoder

Although large language models demonstrate exceptional proficiency in language processing, they face challenges in comprehending the time-evolving patterns inherent in spatio-temporal data. To overcome this limitation, we propose enhancing the capability of large language models to capture temporal dependencies within spatio-temporal contexts. This is accomplished by integrating a spatio-temporal encoder that incorporates a multi-level temporal convolutional network. By doing so, we enable the model to effectively capture the intricate temporal dependencies across various time resolutions, thereby improving its understanding of the complex temporal dynamics present in the spatio-temporal data. Specifically, our spatio-temporal encoder is composed of two key components: a gated dilated convolution layer and a multi-level correlation injection layer. Let's formalize this architecture as:

$$\Psi_r^{(l)} = (\bar{\mathbf{W}}_k^{(l)} * \mathbf{E}_r^{(l)} + \bar{\mathbf{b}}_k^{(l)}) \cdot \delta(\bar{\mathbf{W}}_g^{(l)} * \mathbf{E}_r^{(l)} + \bar{\mathbf{b}}_g^{(l)}) + \mathbf{E}_r'^{(l)} \quad (3)$$

We begin with the initial spatio-temporal embedding, denoted as $\mathbf{E}_r \in \mathbb{R}^{T \times d}$. This embedding is obtained by enhancing the original data \mathbf{X} through a linear layer. To address the issue of gradient vanishing, we utilize a slice of \mathbf{E}_r , denoted as $\mathbf{E}_r' \in \mathbb{R}^{T' \times d}$, which is determined by the size of the dilated convolutional kernel. This slice is employed for performing residual operations. To perform the residual operations, we use 1-D dilated convolution kernels $\bar{\mathbf{W}}_k$ and $\bar{\mathbf{W}}_g \in \mathbb{R}^{T_g \times d_{in} \times d_{out}}$, along with the corresponding bias terms $\bar{\mathbf{b}}_k$ and $\bar{\mathbf{b}}_g \in \mathbb{R}^{d_{out}}$. The sigmoid activation function δ is applied to control the degree of information preservation during the repeated convolution operation. After the gated temporal dilated convolutional layer encoding, we are able to effectively capture the temporal dependencies across multiple time steps, resulting in the generation of temporal representations.

These representations contain different levels of temporal dependencies, reflecting various granularity-aware time-evolving patterns. To preserve these informative patterns, we introduce a multi-level correlation injection layer. This layer is designed to incorporate correlations between different levels and is formalized as:

$$\mathbf{S}_r^{(l)} = (\mathbf{W}_s^{(l)} * \Psi_r^{(l)} + \mathbf{b}_s^{(l)}) + \mathbf{S}_r^{(l-1)} \quad (4)$$

We have the convolution kernel $\mathbf{W}_s \in \mathbb{R}^{T_s \times d_{out} \times d_{out}'}$ and the bias vector $\mathbf{b}_s \in \mathbb{R}^{d_{out}'}$. These are employed after L layers of encoding. A simple non-linear layer is employed to merge the results from equations 3 and 4, and the final spatio-temporal dependency representations are denoted as $\tilde{\Psi} \in \mathbb{R}^{R \times F \times d}$. To address the diverse set of urban scenarios that may arise downstream, our proposed spatio-temporal encoder is designed to be independent of graph structures when modeling spatial correlations. This is particularly crucial because in zero-shot prediction contexts, the spatial relationships between entities may be unknown or difficult to ascertain. By not relying on explicit graph structures, our encoder can effectively handle a broad spectrum of urban scenarios, where spatial correlations and dependencies can vary or be challenging to define in advance. This flexibility enables our model to adapt and perform well, ensuring its applicability in a wide range of urban contexts.

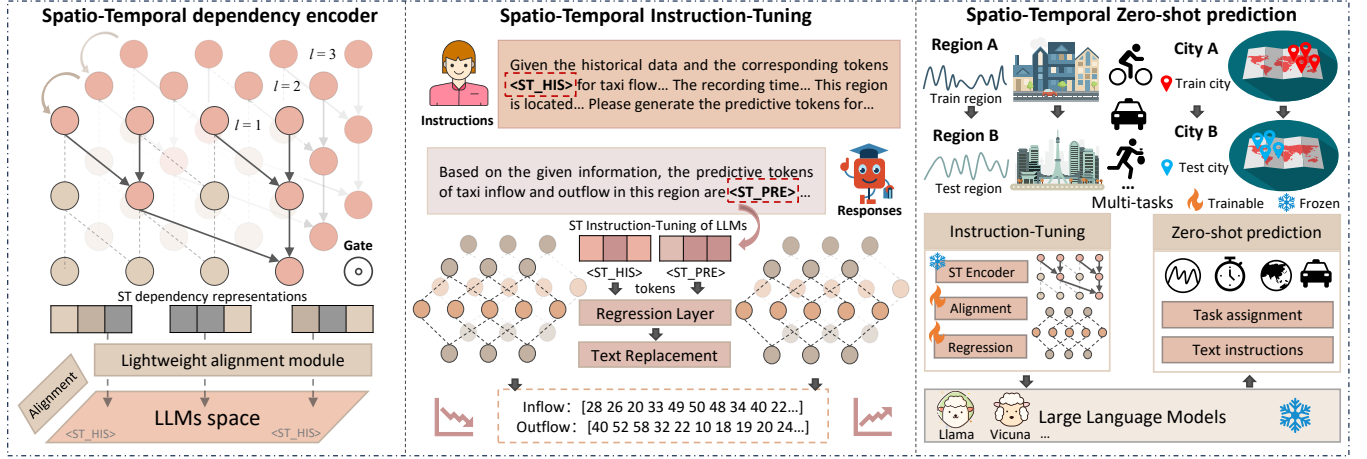


Figure 2: The overall architecture of the proposed spatio-temporal language model UrbanGPT.

3.2 Spatio-Temporal Instruction-Tuning

3.2.1 Spatio-Temporal-Text Alignment. In order to enable language models to effectively comprehend spatio-temporal patterns, it is crucial to align textual and spatio-temporal information. This alignment allows for the fusion of different modalities, resulting in a more informative representation. By integrating contextual features from both textual and spatio-temporal domains, we can capture complementary information and extract higher-level semantic representations that are more expressive and meaningful. To achieve this objective, we utilize a lightweight alignment module to project the spatio-temporal dependencies representations $\tilde{\Psi}$. This projection involves the use of parameters $\mathbf{W}_p \in \mathbb{R}^{d \times d_L}$ and $\mathbf{b}_p \in \mathbb{R}^{d_L}$, where d_L represents the commonly used hidden dimension in language models (LLMs).

The resulting projection, denoted as $\mathbf{H} \in \mathbb{R}^{R \times F \times d_L}$, are represented in the instructions using special tokens as: $\langle \text{ST_start} \rangle$, $\langle \text{ST_HIS} \rangle$, ..., $\langle \text{ST_HIS} \rangle$, $\langle \text{ST_end} \rangle$. Here, $\langle \text{ST_start} \rangle$ and $\langle \text{ST_end} \rangle$ serve as identifiers marking the beginning and end of the spatio-temporal token. These identifiers can be included in the large-scale language model by expanding its vocabulary. The placeholder $\langle \text{ST_HIS} \rangle$ represents the spatio-temporal token and corresponds to the projection \mathbf{H} in the hidden layer. By employing this technique, the model gains the ability to discern spatio-temporal dependencies, thereby enhancing its proficiency in successfully performing spatio-temporal predictive tasks within urban scenes.

3.2.2 Spatio-Temporal Prompt Instructions. In scenarios involving spatio-temporal prediction, both temporal and spatial information contain valuable semantic details that contribute to the model's understanding of spatio-temporal patterns within specific contexts. For instance, traffic flow in the early morning differs significantly from rush hour, and there are variations in traffic patterns between commercial and residential areas. As a result, we recognize the potential of representing both temporal and spatial information as prompt instruction text. We leverage the text understanding capabilities of large language models to encode this information, enabling associative reasoning for downstream tasks.

In our UrbanGPT framework, we integrate multi-granularity time information and spatial details as instruction inputs for the

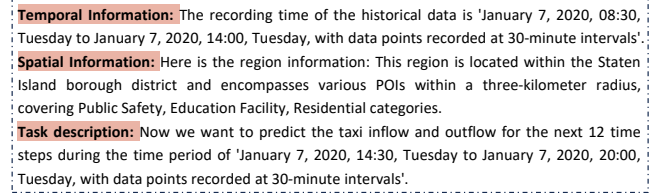


Figure 3: Illustration of spatio-temporal prompt instructions encoding the time- and location-aware information.

large language model. Time information includes factors such as the day of the week and the hour of the day, while regional information encompasses the city, administrative areas, and nearby points of interest (POI) data, among others. By incorporating these diverse elements, UrbanGPT is capable of identifying and assimilating spatio-temporal patterns across different regions and time-frames. This enables the model to encapsulate these insights within complex spatio-temporal contexts, thereby enhancing its ability for zero-shot reasoning. The design of the instructional text for spatio-temporal information is illustrated in Figure 3.

3.2.3 Spatio-Temporal Instruction-Tuning of LLMs. When it comes to incorporating detailed spatio-temporal textual descriptions, the next stage is to fine-tune large language models (LLMs) using instructions to generate spatio-temporal forecasts in textual format. However, this approach poses two challenges. *Firstly*, spatio-temporal forecasting typically relies on numerical data, which differs in structure and patterns from natural language that language models excel at processing, focusing on semantic and syntactic relationships. *Secondly*, large language models are typically pre-trained using a multi-classification loss to predict vocabulary, resulting in a probability distribution of potential outcomes. This contrasts with the continuous value distribution required for regression tasks.

To address these challenges, UrbanGPT adopts a different strategy by refraining from directly predicting future spatio-temporal values. Instead, it generates forecasting tokens that aid in the prediction process. These tokens are subsequently passed through a regression layer, which maps the hidden representations to generate more accurate predictive values. The formulation of the regression

layer in our spatio-temporal instruction-tuning paradigm is:

$$\hat{\mathbf{Y}}_{r,f} = \mathbf{W}_3[\sigma(\mathbf{W}_1\mathbf{H}_{r,f}), \sigma(\mathbf{W}_2\Gamma_{r,f})] \quad (5)$$

The prediction result, denoted as $\hat{\mathbf{Y}}_{r,f} \in \mathbb{R}^P$, is obtained using the rectified linear unit activation function, represented by σ . The hidden representations of the forecasting tokens, denoted as $\Gamma_{r,f} \in \mathbb{R}^{d_L}$, are introduced as a novel term in the vocabulary of large language models (LLMs). The regression layer is formulated using weight matrices $\mathbf{W}_1 \in \mathbb{R}^{d' \times d_L}$, $\mathbf{W}_2 \in \mathbb{R}^{d' \times d_L}$, and $\mathbf{W}_3 \in \mathbb{R}^{P \times 2d'}$, where $[\cdot, \cdot]$ represents the concatenation operation. While the probability distribution of the forecasting tokens remains relatively stable, their hidden representations contain rich spatio-temporal contextual attributes that capture dynamic spatio-temporal interdependencies. This enables our model to provide precise predictions by leveraging this contextual information.

3.3 Model Optimization

Building upon the baseline model [1, 8], we adopt the absolute error loss as our regression loss function. This choice allows us to effectively handle predictions across a wide range of urban scenarios. Additionally, we introduce a classification loss as a joint loss to cater to diverse task requirements. To ensure optimal performance, our model optimizes different losses based on the specific task inputs. For instance, we utilize the regression loss for tasks such as traffic flow prediction, while employing the classification loss for tasks like crime prediction. This approach enables our model to effectively address the unique challenges posed by each task and deliver accurate predictions in various urban scenarios.

$$\begin{aligned} \mathcal{L}_c &= -\frac{1}{N} \sum_{i=1}^N [\delta(y_i) \cdot \log(\hat{y}_i) + (1 - \delta(y_i)) \cdot \log(1 - \hat{y}_i)] \\ \mathcal{L}_r &= \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|; \quad \mathcal{L} = \mathcal{L}_{LLMs} + \mathcal{L}_r + \mathcal{L}_c \end{aligned} \quad (6)$$

Here, y_i represents a sample from $\hat{\mathbf{Y}}$, and N represents the total number of samples, which is calculated as the product of R , T , and F . We use various loss functions in our model, including \mathcal{L}_c for binary cross-entropy loss, \mathcal{L}_r for regression loss, and the cross-entropy loss adopted in our spatio-temporal language models. To capture probability distributions from the prediction, we employ the sigmoid function denoted by δ . Each of these loss functions plays a specific role in our model, allowing us to effectively handle classification, regression, and language modeling tasks as needed.

4 Evaluation

In this section, we aim to assess the capabilities of our proposed model across various settings by addressing five key questions:

- **RQ1:** What is the performance and generalization capability of UrbanGPT in diverse zero-shot spatio-temporal prediction tasks?
- **RQ2:** How does UrbanGPT perform when compared to existing spatio-temporal models in classical supervised scenarios?
- **RQ3:** What specific contributions do the proposed key components bring to enhance the capabilities of our UrbanGPT model?
- **RQ4:** Can the proposed model robustly handle the forecasting scenarios with varying spatio-temporal patterns?
- **RQ5:** How does the performance of UrbanGPT compare to existing large language models in zero-shot learning scenarios?

4.1 Experimental Setting

4.1.1 Dataset Description. To evaluate the effectiveness of the proposed model in predicting spatio-temporal patterns across various urban computing scenarios, we conducted experiments using four distinct datasets: NYC-taxi, NYC-bike, NYC-crime, and CHI-taxi. These datasets encompass a wide range of data sources to capture the dynamic nature of urban environments, including records of taxi travel, bike trajectories, crime incidents in New York City, and taxi travel data in Chicago. To facilitate our analysis, we partitioned the cities into grid-like regions based on latitude and longitude information. Within specific time intervals, we aggregated statistical measures for each region. For example, this involved calculating the number of taxi inflows and outflows within a 30-minute period in region A, or determining the count of theft incidents within a day in region B. Furthermore, Points of Interest (POIs) data can be obtained through APIs provided by map services, utilizing the latitude and longitude of different regions. For more comprehensive data descriptions, please refer to the Appendix.

4.1.2 Hyperparameters Settings. During the instruction-tuning phase, we maintained consistency with Vicuna’s default settings for the majority of hyperparameters. These include a learning rate of 0.002, using the AdamW optimizer, and a warmup rate of 0.03. To improve training efficiency, we set the batch size to 4. The parameters for the dilation convolution kernel in the time encoder are set as follows: d_{in} , d_{out} , and d'_{out} are all set to 32, with a dilation factor of 1. For our prediction task, we aim to predict the next 12 steps of data based on the previous 12 steps. Both the history length (H) and prediction length (P) are set to 12. The projection layer parameters are configured with d set to 64 and d_L set to 4096. Lastly, the hidden layer parameter d' for the regression layer is set to 128. We generated a total of 863,040 training samples from the NYC-taxi, NYC-bike, and NYC-crime datasets for instruction-tuning. The training process was conducted on 8 × NVIDIA A100-SXM4-40GB GPUs and took approximately 16 hours to complete.

4.1.3 Evaluation Protocols. In order to investigate the capabilities of large language models in analyzing diverse spatio-temporal data across different regions, we selected a subset of taxi, bike, and crime data from various areas of New York City as our training set.

- **Zero-Shot Learning Scenarios.** We assessed the model performance by predicting future spatio-temporal data from regions in NYC or even Chicago that were unseen in the training phase.
- **Supervised Learning Scenarios.** We evaluated the model using future data from the same regions as the training set.

For regression tasks, we maintained a consistent training and testing methodology across all baseline models. When it came to classification tasks involving crime data, we utilized binary cross-entropy as the loss function for training and testing the models. Our experiments were conducted using the robust vicuna-7b [52] as the foundational large language model for UrbanGPT. For a more comprehensive understanding of our methodology and experimental setup, please refer to the appendix for detailed information.

Table 1: Our model’s performance in zero-shot prediction is evaluated on three diverse datasets: NYC-taxi, NYC-bike, and NYC-crime, providing a comprehensive assessment of its predictive capabilities in unseen situations.

Model	Dataset	NYC-taxi				NYC-bike				NYC-crime			
	Type	Inflow		Outflow		Inflow		Outflow		Burglary		Robbery	
	Metrics	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	Macro-F1	Recall	Macro-F1	Recall
AGCRN		10.86	26.51	13.15	36.45	3.41	7.98	3.42	8.08	0.48	0.00	0.49	0.01
ASTGCN		9.75	24.12	12.42	33.28	5.58	11.58	5.78	12.29	0.49	0.01	0.55	0.09
GWN		10.73	26.50	9.67	26.74	3.32	8.17	3.07	7.52	0.48	0.00	0.52	0.04
MTGNN		10.16	25.84	12.59	35.38	3.18	7.62	3.20	7.65	0.64	0.27	0.65	0.30
STWA		11.28	28.97	13.54	38.61	4.59	10.94	4.35	10.67	0.48	0.00	0.51	0.03
STSGCN		18.97	41.38	20.07	45.79	6.85	14.98	6.54	14.77	0.48	0.00	0.48	0.00
STGCN		12.54	30.80	14.32	39.58	4.11	9.21	4.45	9.62	0.48	0.00	0.64	0.30
TGCN		10.04	25.10	10.98	30.03	2.88	6.55	2.91	6.42	0.56	0.10	0.58	0.13
DMVSTNET		11.00	28.29	10.59	29.20	3.80	9.87	3.65	9.21	0.48	0.01	0.59	0.15
ST-LSTM		16.97	34.43	18.93	44.10	7.78	15.41	6.92	17.12	0.48	0.00	0.49	0.03
GPT4TS		9.72	24.51	10.85	31.00	3.16	7.45	3.23	7.53	0.48	0.00	0.49	0.02
<i>UrbanGPT</i>		6.16	16.92	6.83	21.78	2.02	5.16	2.01	5.03	0.67	0.34	0.69	0.42

4.1.4 Evaluation Metrics. For regression tasks, we employed MAE (Mean Absolute Error) and RMSE (Root Mean Square Error) as evaluation metrics. These metrics quantify the discrepancies between the predicted outcomes and the actual labels, with lower values indicating superior performance [12, 51]. In the case of classification tasks, we utilized Recall and Macro-F1 as evaluation metrics to assess performance. Recall measures the model’s ability to correctly identify positive instances, while Macro-F1 is a comprehensive performance metric that combines precision and recall to provide an overall measure of classification accuracy [11, 31].

4.1.5 Baseline Model. To establish a comprehensive baseline for our proposed method, we conducted a thorough comparison against 11 advanced models, which we have categorized into five distinct groups: traditional methods, RNN-based models, attention-based models, graph neural network (GNN)-based models, and large language model-based methods. We will provide detailed descriptions of each baseline category below.

RNN-based Spatio-Temporal Methods:

- **ST-LSTM**[34]: It incorporates the Long Short-Term Memory to capture temporal dependencies in the spatio-temporal data.
- **AGCRN**[1]: RNNs are employed to capture temporal correlations, allowing for the representation of the evolving patterns over time.
- **DMVSTNET**[41]: In this method, RNNs are utilized to effectively model temporal dependencies, capturing the patterns that evolve over time. Furthermore, convolutional networks and fully connected layers are employed to capture local spatial correlations and establish meaningful spatial relationships.

Attention-based Spatio-Temporal Approaches:

- **ASTGCN**[8]: In this method, attention mechanisms are employed to capture multi-granularity temporal correlation features.
- **STWA**[5]: The model incorporates personalized temporal and spatial parameters into the attention module, allowing for the modeling of dynamic spatio-temporal correlations.

Spatio-Temporal GNNs:

- **GWN**[39]: It incorporates a learnable graph structure and 1-D convolutions to effectively learn spatio-temporal dependencies.
- **MTGNN**[38]: It utilizes a learnable graph structure to model multivariate temporal correlations. MTGNN employs 1-D dilation convolutions to generate temporal representations.
- **TGCN**[49]: This model combines graph neural networks (GNNs) for spatial correlation modeling and recurrent neural networks (RNNs) for temporal correlation modeling.
- **STGCN**[44]: It uses gated temporal convolutions and GNNs to model temporal and spatial dependencies, respectively.
- **STSGCN**[27]: It introduces a spatio-temporal graph construction to learn spatial correlations across adjacent time steps.

Large language models-based methods:

- **GPT4TS**[54]: This model utilizes a pre-trained large language model, GPT-2, as its backbone to predict multivariate time series.

4.2 Zero-Shot Prediction Performance (RQ1)

In this section, we thoroughly evaluate the predictive performance of our proposed model in zero-shot scenarios. The results of our evaluation are presented in Table 1 and visualized in Figure 4. Our objective is to assess the model’s effectiveness in predicting spatio-temporal patterns in geographical areas that it has not encountered during training. This evaluation encompasses both cross-region and cross-city settings, allowing us to gain insights into the model’s generalization capabilities across different locations.

4.2.1 Prediction on Unseen Regions within a City. Cross-region scenarios entail using data from certain regions within a city to forecast future conditions in other regions that have not been encountered by the model. Through a thorough analysis of the model’s performance in these cross-region predictions, we can draw attention to three significant observations:

i) Superior Zero-shot Predictive Performance. The results presented in Table 1 highlight the exceptional performance of our

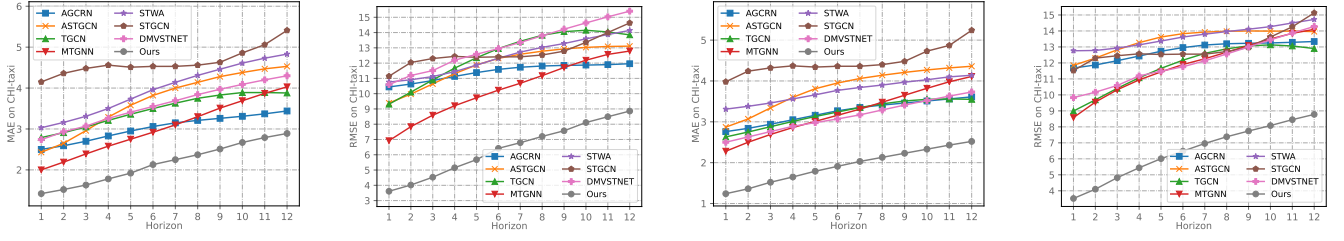


Figure 4: Time step-based prediction comparison experiment conducted on the CHI-taxi dataset.

proposed model in both regression and classification tasks on various datasets, surpassing the baseline models in zero-shot prediction. The success of our model can be attributed to two key factors.

- **Spatio-Temporal-Text-Alignment.** The alignment of spatio-temporal contextual signals with the text comprehension abilities of language models plays a pivotal role in the success of our proposed model. This fusion enables the model to effectively capitalize on both the encoded urban dynamics from the spatiotemporal signals and the comprehensive understanding of textual context provided by the LLMs. By leveraging these two essential aspects, our model achieves the remarkable ability to generalize its prediction capabilities in zero-shot scenarios.
- **Spatio-Temporal Instruction-Tuning.** This adaptive tuning process empowers the LLM to effectively integrate crucial information from the instructions, enhancing its comprehension of the complex relationships and dependencies between spatial and temporal factors. By seamlessly merging the spatio-temporal instruction-tuning with the spatio-temporal dependency encoder, our proposed model, UrbanGPT, successfully preserves universal and transferable spatio-temporal knowledge. Consequently, the model becomes capable of capturing the fundamental patterns and dynamics that govern spatio-temporal phenomena, enabling it to make precise predictions in downstream zero-shot scenarios.

ii) Enhanced Urban Semantic Understanding. Urban semantics offer valuable insights into the diverse dimensions of spatial and temporal characteristics. Our approach involves training our model on a wide range of datasets, enriching its understanding of spatio-temporal dynamics across different timeframes and geographical locations. In contrast, baseline models tend to prioritize encoding temporal and spatial dependencies, neglecting the nuanced semantics that differentiate regions, timeframes, and data categories. By incorporating comprehensive semantic awareness into our UrbanGPT, we significantly enhance its ability to make accurate zero-shot predictions in previously unseen regions.

iii) Improved Performance in Sparse Data Scenarios. Predicting spatio-temporal patterns in sparse data environments is challenging as models tend to overfit when data points are scarce. This challenge is particularly notable when predicting crimes, where data is often sparse but crucial for accurate predictions. Baseline models struggle in cross-regional prediction tasks under these sparse conditions, resulting in low recall scores that indicate potential overfitting. To overcome this limitation, our model integrates spatio-temporal learning with large language models (LLMs) using an effective spatio-temporal instruction-tuning paradigm. By incorporating rich semantic insights, our approach enhances the model's

spatio-temporal representations, enabling it to effectively handle sparse data and achieve improved prediction accuracy.

4.2.2 Cross-City Prediction Task. To assess the performance of our model in cross-city prediction tasks, we conducted tests on the CHI-taxi dataset, which was not seen during the training phase. The results, depicted in Figure 4, yielded the following observations:

- **Consistency in Multi-step Prediction:** Our model consistently outperforms the comparison method at each time step. Notably, it maintains a significant advantage in both short-term and long-term spatio-temporal prediction, demonstrating the robustness of our proposed model in cross-city prediction scenarios.
- **Effective Knowledge Transfer Across Cities:** The prediction results obtained from the CHI-taxi dataset validate the superior forecasting capabilities of our model in cross-city scenarios. This enhancement can be attributed to the integration of spatio-temporal encoders with the spatio-temporal instruction-tuning paradigm. By incorporating these components, our model effectively captures universal and transferable spatio-temporal patterns, allowing it to make accurate predictions. Additionally, by considering different geographical information and temporal factors alongside the learned transferred knowledge, our model successfully associates spatio-temporal patterns exhibited by similar functional areas and historical periods. This comprehensive understanding provides valuable insights for making precise zero-shot predictions in cross-city scenarios.

4.3 Classical Supervised Prediction Task (RQ2)

This section examines the predictive capabilities of our UrbanGPT in end-to-end supervised prediction scenarios, as presented in Table 2. We will discuss the results from two perspectives below.

Enhanced Long-Term Forecasting Abilities: We examined the model's effectiveness in long-term spatio-temporal forecasting by utilizing test datasets that spanned broader time intervals. For instance, we trained the model using data from 2017 and evaluated its performance on data from 2021. The results of these experiments demonstrate that our UrbanGPT possesses a significant advantage over the baselines, highlighting its superior ability to generalize across different temporal landscapes. This capability reduces the need for frequent retraining or incremental updates, making the model more aligned with real-world applications. Additionally, the experiments have confirmed that incorporating additional textual knowledge does not hinder model performance or introduce noise, thus further validating the feasibility of utilizing large language models to enhance spatio-temporal forecasting tasks.

Table 2: Evaluation of performance in the end-to-end supervised setting on the NYC-taxi and NYC-bike datasets.

Model	NYC-taxi				NYC-bike			
	Inflow		Outflow		Inflow		Outflow	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
AGCRN	2.83	8.35	2.62	9.21	3.30	7.65	3.38	7.73
ASTGCN	5.41	18.04	5.00	19.29	3.87	7.93	3.66	7.69
GWN	3.91	11.93	2.89	10.85	4.30	9.04	3.88	8.29
MTGNN	3.09	10.13	2.61	10.96	3.31	7.47	3.26	7.61
STWA	3.90	12.64	3.15	11.32	4.23	9.07	4.18	9.18
STSGCN	4.57	13.93	4.41	15.87	5.10	12.23	4.72	10.78
STGCN	3.45	9.82	3.17	10.53	3.88	9.23	3.90	9.08
TGCN	3.99	11.47	3.31	11.58	4.12	7.92	4.11	7.84
DMVSTNET	3.83	11.55	2.76	9.88	3.71	7.95	3.69	7.92
ST-LSTM	7.78	15.41	6.92	17.12	5.00	11.52	4.96	11.41
<i>UrbanGPT</i>	2.50	6.78	1.71	6.68	3.11	7.10	3.01	6.94

Spatial Semantic Understanding: Accurately capturing spatial correlations is crucial in the realm of spatio-temporal prediction. Traditional methods often employ graph networks or attention mechanisms to analyze these correlations. Models lacking dedicated spatial correlation modules, like LSTM, tend to underperform as they overlook the spatial context. In contrast, our model compensates for the absence of explicit spatial encoders by integrating extensive geographic and points of interest (POIs) data within the textual input. This approach enables the model to comprehend the shared characteristics of areas with similar functions at a higher semantic level. Consequently, it deduces patterns of correlation between various functional zones and effectively represents the interconnections among different regions.

4.4 Ablation Study (RQ3)

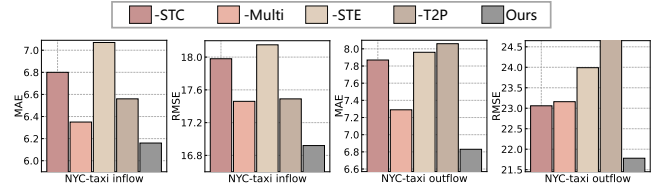
This section investigates the impact of different key components on the performance of our model, as illustrated in Figure 5. Our rigorous testing primarily revolves around the zero-shot scenario using the NYC-taxi dataset. Through our analysis, we have distilled the benefits offered by the different modules into four key points.

- **(1) Impact of Spatial and Temporal Context: -STC.** By removing time and spatial information from the instruction text, we observed a noticeable decline in the model’s performance.

This can be attributed to the lack of temporal information, forcing the model to rely solely on spatio-temporal encoders for encoding time-related features and performing prediction tasks. Furthermore, the absence of spatial information hindered the model’s ability to capture spatial correlations, making it challenging to analyze the distinct spatio-temporal patterns of different areas.

- **(2) Impact of Instruction-Tuning with Diverse Datasets: -Multi.** We conducted our training solely on the NYC-taxi data to examine whether incorporating multiple datasets would provide valuable insights to the LLMs in zero-shot scenarios.

However, this restricted training approach limited the model’s ability to fully uncover the spatio-temporal dynamics of cities due to the absence of diverse urban indicators. As a result, the model’s performance was suboptimal. By integrating diverse spatio-temporal data from multiple sources, our model can effectively capture the unique characteristics of different geographical locations and their evolving spatio-temporal patterns.

**Figure 5: Ablation study of our proposed UrbanGPT.**

- **(3) Impact of Spatio-Temporal Encoder: -STE.** In this variant, we disable the spatio-temporal encoder to investigate its effect on aligning the large language model with encoded urban dependency dynamics into latent embedding space.

The results clearly indicate that the absence of the spatio-temporal encoder significantly hampers the performance of the large language model in spatio-temporal prediction scenarios. This underscores the crucial role played by the proposed spatio-temporal encoder in enhancing the model’s predictive capabilities.

- **(4) Regression Layer Incorporation in Instruction-Tuning: T2P.** We explicitly instructed UrbanGPT to generate its predictions in a textual format. However, the suboptimal performance indicates the challenges in utilizing LLMs for precise numerical regression tasks, as opposed to employing regression layers.

The primary challenge stems from the model’s dependence on multi-class loss for optimization during training, leading to a mismatch between the model’s probabilistic output and the continuous value distribution necessary for spatio-temporal forecasting. To bridge this gap, we incorporated a regression predictor into our model, significantly improving its capacity to generate more precise numerical predictions for regression tasks.

4.5 Model Robustness Study (RQ4)

In this section, we focus on evaluating the robustness of our UrbanGPT across different spatio-temporal pattern scenarios. We categorize regions based on the magnitude of numerical variations, such as taxi flow, during a specific time period. Lower variance indicates stable temporal patterns, while higher variance suggests diverse spatio-temporal patterns in active commercial zones or densely populated areas. Our findings, shown in Figure 6, reveal that most models perform well in regions with lower variance, where patterns remain relatively stable. However, the baseline model struggles in regions with high variance, particularly within the (0.75, 1.0] range, resulting in inaccurate predictions. This limitation may stem from the baseline model’s difficulty in inferring spatio-temporal patterns in unseen regions during zero-shot scenarios. In practical applications, accurate prediction of densely populated or bustling areas is crucial for urban governance, such as traffic light control and security scheduling. Our UrbanGPT demonstrates significant performance improvement in the (0.75, 1.0] interval, highlighting its powerful zero-shot prediction capability with our method.

4.6 Case Study

In our case study, we thoroughly evaluate several large language models (LLMs) for zero-shot spatio-temporal prediction. We emphasize the challenges these models face in directly understanding

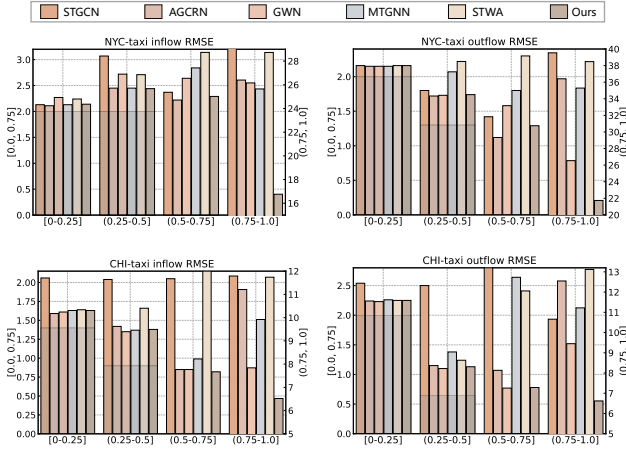


Figure 6: Robustness study of the UrbanGPT model.

spatio-temporal patterns from numeric geo-series data. In contrast, we showcase the exceptional performance of our proposed UrbanGPT framework in capturing universal spatio-temporal patterns and its ability to generalize effectively across various zero-shot spatio-temporal forecasting scenarios. For a more comprehensive understanding, please refer to the Appendix.

5 Related Work

Deep Spatio-temporal Prediction Models. Deep spatio-temporal prediction methods have gained prominence in deep learning due to their impressive performance. These models typically consist of two components: temporal dependency modeling and spatial correlation encoding. Early models like D-LSTM [45] and ST-resnet [47] used RNNs and convolutional networks to model temporal and spatial dependencies. Graph neural networks (GNNs) proved to be a natural fit for spatial correlation modeling, as seen in models like STGCN [44] and DCRNN [14], which utilized graph structures based on node distances. Techniques such as learnable region-wise graph structures [38, 39] and dynamic spatio-temporal graph networks [9, 50] have further enhanced spatial correlation modeling.

Furthermore, researchers have explored approaches such as multi-scale temporal learning [35] and multi-granularity temporal learning [8] to encode temporal dependencies. These strategies enable the capture of features like long-term and short-term correlations as well as periodicity. These advancements have significantly contributed to the progress of spatio-temporal prediction. However, it is worth noting that the majority of these studies are tailored for supervised contexts, with limited research and development focused on zero-shot spatio-temporal forecasting. This represents an important area that requires further exploration.

Spatio-Temporal Pre-training Techniques have recently received significant research attention. These techniques primarily focus on generative [16, 17, 25] and contrastive [48] pre-training models to enhance the predictive performance of downstream tasks. There has also been extensive exploration of pretrain-finetune frameworks for few-shot learning scenarios [13, 20], aiming to improve knowledge transferability by aligning source and target data. However, these approaches require training or fine-tuning on

target data and lack zero-shot prediction capabilities. In this work, we address the challenge of data scarcity in downstream urban scenarios by proposing UrbanGPT. Our model demonstrates the ability to generalize well across various scenarios, mitigating the need for extensive training or fine-tuning on target data.

Large Language Models. The emergence of large language models [3, 22] has recently attracted significant attention due to their unprecedented machine performance in tasks like text understanding and reasoning. These models have become a hot topic, demonstrating the potential to advance from intelligent algorithms to artificial intelligence. Open-source large language models such as Llama [29, 30], Vicuna [52], and ChatGLM [46] have been released, leading to researchers exploring their application in various fields to enhance transfer learning capabilities with domain knowledge from these models. In the computer vision domain, researchers have combined multimodal large language models with prompt learning methods to achieve zero-shot predictions in downstream tasks [26, 53, 55]. Furthermore, the capabilities of LLMs in graph reasoning [2, 4, 28], recommendation [10, 24, 36] and traffic analysis [6] have been extensively studied. However, the utilization of large language models for zero-shot spatio-temporal prediction tasks in the field of urban intelligence remains largely unexplored.

6 Conclusion

We present the UrbanGPT, a spatio-temporal large language model, to generalize well in diverse urban scenarios. To seamlessly align the spatio-temporal contextual signals with LLMs, we introduce a spatio-temporal instruction-tuning paradigm. This empowers the UrbanGPT with the remarkable ability to learn universal and transferable spatio-temporal patterns across various types of urban data. Through extensive experiments and meticulous ablation studies, we demonstrate the exceptional effectiveness of the UrbanGPT’s architecture and its key components. This work is among the pioneering efforts to explore the potential of large language models for smart cities, and it demonstrates the exceptional effectiveness of our proposed spatio-temporal LLM. However, it is important to acknowledge that while the results are promising, there are still limitations to be addressed in future studies. As a first step, we are actively engaged in collecting a more diverse range of urban data to enhance and refine the capabilities of our UrbanGPT across a broader spectrum of urban computing domains. Additionally, understanding the decision-making process of our UrbanGPT is of importance. While the model demonstrates exceptional performance, providing interpretability and explainability is equally essential. Future research efforts will focus on empowering our model with the ability to interpret and explain its predictions.

Acknowledgments

This work was partially supported by grants from the National Natural Science Foundation of China (No. 62072188), the National Foreign Expert Project of China’s Ministry of Science and Technology (No. G2023163015L), and the Guangzhou Municipal Science and Technology Plan Project (No. 2024B01W0007). We are also grateful for the support provided by the HKU Musketeers Foundation Institute of Data Science & Department of Computer Science, PAZHOU LAB, and the Guangdong Provincial Key Laboratory of Multimodal Big Data Intelligent Analysis.

References

- [1] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. In *NeurIPS*. 17804–17815.
- [2] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, et al. 2024. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. (2024), 17682–17690.
- [3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, et al. 2020. Language Models Are Few-Shot Learners. In *NeurIPS*. 1877–1901.
- [4] Ziwei Chai, Tianjie Zhang, Liang Wu, Kaiqiao Han, Xiaohai Hu, Xuanwen Huang, and Yang Yang. 2023. GraphLLM: Boosting Graph Reasoning Ability of Large Language Model. arXiv:2310.05845
- [5] Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. 2022. Towards Spatio-Temporal Aware Traffic Time Series Forecasting. In *ICDE*. 2900–2913.
- [6] Longchao Da, Kuanru Liou, Tiejun Chen, Xuesong Zhou, Xiangyong Luo, Yezhou Yang, and Hua Wei. 2023. Open-TI: Open Traffic Intelligence with Augmented Language Model. arXiv:2401.00211
- [7] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *WWW*. 1459–1468.
- [8] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. In *AAAI*. 922–929.
- [9] Liangzhe Han, Bowen Du, Leilei Sun, Yanjie Fu, Yisheng Lv, and Hui Xiong. 2021. Dynamic and Multi-Faceted Spatio-Temporal Deep Learning for Traffic Speed Forecasting. In *KDD*. 547–555.
- [10] Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging large language models for sequential recommendation. In *Recsys*. 1096–1102.
- [11] Chao Huang, Junbo Zhang, Yu Zheng, and Nitesh V Chawla. 2018. DeepCrime: Attentive hierarchical recurrent networks for crime prediction. In *CIKM*. 1423–1432.
- [12] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. 2023. PDFormer: Propagation Delay-Aware Dynamic Long-Range Transformer for Traffic Flow Prediction. In *AAAI*. 4365–4373.
- [13] Yilun Jin, Kai Chen, and Qiang Yang. 2022. Selective cross-city transfer learning for traffic prediction via source city region re-weighting. In *KDD*. 731–741.
- [14] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In *ICLR*.
- [15] Zhonghang Li, Chao Huang, Lianghao Xia, Yong Xu, and Jian Pei. 2022. Spatial-Temporal Hypergraph Self-Supervised Learning for Crime Prediction. In *ICDE*. 2984–2996.
- [16] Zhonghang Li, Lianghao Xia, Yong Xu, and Chao Huang. 2023. GPT-ST: Generative Pre-Training of Spatio-Temporal Graph Neural Networks. In *Advances in Neural Information Processing Systems*. 70229–70246.
- [17] Zhonghang Li, Lianghao Xia, Yong Xu, and Chao Huang. 2024. FlashST: A Simple and Universal Prompt-Tuning Framework for Traffic Prediction. arXiv:2405.17898
- [18] Yuxuan Liang, Kun Ouyang, Lin Jing, Sijie Ruan, Ye Liu, Junbo Zhang, David S Rosenblum, and Yu Zheng. 2019. Urbanfm: Inferring fine-grained urban flows. In *KDD*. 3132–3142.
- [19] Binbing Liao, Jingqing Zhang, Chao Wu, Douglas Mclwraith, Tong Chen, Shengwen Yang, Yike Guo, and Fei Wu. 2018. Deep sequence learning with auxiliary information for traffic prediction. In *KDD*. 537–546.
- [20] Bin Lu, Xiaoying Gan, Weinan Zhang, Huaxiu Yao, Luoyi Fu, and Xinbing Wang. 2022. Spatio-Temporal Graph Few-Shot Learning with Cross-City Knowledge Transfer. In *KDD*. 1162–1172.
- [21] Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. 2021. A survey on deep learning for human mobility. *ACM Computing Surveys (CSUR)* 55, 1 (2021), 1–44.
- [22] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, et al. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*. 27730–27744.
- [23] Zheyi Pan, Yuxuan Liang, Weifeng Wang, et al. 2019. Urban Traffic Prediction from Spatio-Temporal Data Using Deep Meta Learning. In *KDD*.
- [24] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation Learning with Large Language Models for Recommendation. In *WWW*. 3464–3475.
- [25] Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. 2022. Pre-training Enhanced Spatial-temporal Graph Neural Network for Multivariate Time Series Forecasting. In *KDD*. 1567–1577.
- [26] Sheng Shen, Shijia Yang, Tianjun Zhang, Bohan Zhai, Joseph E. Gonzalez, Kurt Keutzer, and Trevor Darrell. 2024. Multitask Vision-Language Prompt Tuning. In *WACV*. 5656–5667.
- [27] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2020. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. In *AAAI*. 914–921.
- [28] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2024. GraphGPT: Graph Instruction Tuning for Large Language Models. arXiv:2310.13023
- [29] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, et al. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971
- [30] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288
- [31] Beibei Wang, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2021. GSNet: Learning Spatial-Temporal Correlations from Geographical and Semantic Aspects for Traffic Accident Risk Forecasting. *AAAI*, 4402–4409.
- [32] Binwu Wang, Yudong Zhang, Xu Wang, Pengkun Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. 2023. Pattern expansion and consolidation on evolving graphs for continual traffic prediction. In *KDD*. 2223–2232.
- [33] Hongjian Wang, Daniel Kifer, Corina Graif, and Zhenhui Li. 2016. Crime rate inference with big data. In *KDD*. 635–644.
- [34] Jingyuan Wang, Jiawei Jiang, Wenjun Jiang, Chao Li, and Wayne Xin Zhao. 2021. LibCity: An Open Library for Traffic Prediction. In *SIGSPATIAL*. 145–148.
- [35] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. 2020. Traffic Flow Prediction via Spatial Temporal Graph Neural Network. In *WWW*. 1082–1092.
- [36] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. LLMRec: Large Language Models with Graph Augmentation for Recommendation. arXiv:2311.00423
- [37] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *ICLR*.
- [38] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *KDD*. 753–763.
- [39] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph wavenet for deep spatial-temporal graph modeling. In *IJCAI*. 1907–1913.
- [40] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. 2019. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *WWW*. 2181–2191.
- [41] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, Didi Chuxing, and Zhenhui Li. 2018. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. In *AAAI*.
- [42] Junchen Ye, Leilei Sun, Bowen Du, Yanjie Fu, and Hui Xiong. 2021. Coupled Layer-wise Graph Convolution for Transportation Demand Prediction. In *AAAI*. 4617–4625.
- [43] Xiwen Yi, Yu Zheng, Junbo Zhang, and Tianrui Li. 2016. ST-MVL: filling missing values in geo-sensory time series data. In *IJCAI*. 2704–2710.
- [44] Bing Yu, Haoteng Yin, et al. 2018. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *IJCAI*. 3634–3640.
- [45] Rose Yu, Yaguang Li, Cyrus Shahabi, Ugur Demiryurek, and Yan Liu. 2017. Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting. In *SDM*. 777–785.
- [46] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2023. GLM-130B: An Open Bilingual Pre-trained Model. In *ICLR*.
- [47] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In *AAAI*.
- [48] Qianru Zhang, Chao Huang, Lianghao Xia, Zheng Wang, Zhonghang Li, and Siuming Yiu. 2023. Automated Spatio-Temporal Graph Contrastive Learning. In *WWW*. 295–305.
- [49] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2020. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems (TITS)* 21, 9 (2020), 3848–3858.
- [50] Yusheng Zhao, Xiao Luo, Wei Ju, Chong Chen, Xian-Sheng Hua, and Ming Zhang. 2023. Dynamic Hypergraph Structure Learning for Traffic Flow Forecasting. In *ICDE*. 2303–2316.
- [51] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. GMAN: A Graph Multi-Attention Network for Traffic Prediction. In *AAAI*. 1234–1241.
- [52] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, et al. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. arXiv:2306.05685
- [53] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)* 130, 9 (2022), 2337–2348.
- [54] Tian Zhou, Peisong Niu, xue wang, Liang Sun, and Rong Jin. 2023. One Fits All: Power General Time Series Analysis by Pretrained LM. In *NeurIPS*. 43322–43355.
- [55] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. arXiv:2304.10592

A Appendix

In the appendix section, we provide comprehensive details on the dataset information and experimental settings for both the instruction-tuning phase and the test phase. In addition, we conducted further experiments to explore the long-term forecasting performance of our UrbanGPT and its robustness under different large language model (LLM) backbones. Furthermore, we present a case study demonstrating the effectiveness of our spatio-temporal LLM for zero-shot spatio-temporal predictions.

A.1 Experimental Details Description

A.1.1 Dataset Details. We collected data on taxi flows, bicycle flows, and crime incidents in New York City for training and evaluation. The NYC-taxi dataset contains 263 regions, with each region measuring approximately 3km x 3km. The time sampling interval for this dataset is 30 minutes. The NYC-bike and NYC-crime datasets consist of 2162 regions, each represented by a 1km x 1km grid. The time sampling interval for NYC-bike is also 30 minutes, while for NYC-crime, it is 1 day. All datasets cover the time period from Jan 1, 2016, to Dec 31, 2021, in NYC. The CHI-taxi dataset includes 77 regions, with each region measuring approximately 4km x 4km. This dataset includes all taxi data from January 1, 2021, to December 31, 2021, with a time sampling interval of 30 minutes.

A.1.2 Further Experimental Setup Descriptions. During the instruction-tuning phase, we randomly selected 80 regions from the three datasets in New York City as training data. It's important to note that the region indices were kept consistent for the NYC-bike and NYC-crime datasets. The training sets had specific time intervals: for NYC-taxi datasets, it ranged from January 1, 2017, to March 31, 2017; for NYC-bike datasets, it covered April 1, 2017, to June 30, 2017; and for NYC-crime datasets, it spanned from January 1, 2016, to December 31, 2018. For the pretraining of the spatio-temporal dependency encoder and the baseline models, we utilized the same data for training to optimize the parameters. The maximum epoch count was set at 100. In the testing phase, we conducted the following evaluations:

- (i) **Zero-Shot Prediction.** We also selected an additional 80 regions from the New York City datasets as unseen test data. For the NYC-bike and NYC-taxi datasets, we used the first two weeks of data in 2020 for testing. As for the NYC-crime dataset, we used the entire year of 2020 for testing. In the case of the Chicago city dataset, we evaluated the model using all the data from Dec 2021.
- (ii) **Classical Supervised Prediction:** For evaluation purposes, we chose the data with the longest time interval, which involved testing the model using all the data from the NYC-bike and NYC-taxi datasets specifically for the month of December 2021.

A.2 Long-term Forecasting Performance

In this section, we primarily explored the long-term forecasting capabilities of the UrbanGPT. Specifically, we set both the historical (H) and forecast steps (P) to 48. We selected the competitive models TimesNet [37] and GPT4TS [54] as baselines for the long-term forecasting task. TimesNet implements its temporal representation modeling using a two-dimensional convolutional Inception block after decomposing the time series, while GPT4TS uses the large language model GPT-2 as its backbone. The experimental results under a zero-shot setting on the NYC-taxi and NYC-bike datasets are

shown in Table 3, and the performance advantage effectively showcases the capability of UrbanGPT to extract valuable insights from both long-term and short-term data, resulting in a more accurate modeling of the complex spatio-temporal correlations.

Table 3: Evaluating the long-term forecasting capabilities of our UrbanGPT on Taxi and Bike datasets.

Model	NYC-taxi				NYC-bike			
	Inflow		Outflow		Inflow		Outflow	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
TimesNet	9.37	24.24	10.37	29.78	3.09	7.35	3.19	7.53
GPT4TS	9.72	24.51	10.85	31.00	3.16	7.45	3.23	7.53
<i>UrbanGPT</i>	6.16	16.92	6.83	21.78	2.02	5.16	2.01	5.03

A.3 Exploration with Different LLMs Backbones

To further explore the impact of different LLMs as backbones on the UrbanGPT, we added an additional variant, Llama-2-7b, for retraining and testing. The experimental results for the zero-shot settings on the NYC-taxi and NYC-bike datasets are presented in Table 4. The findings indicate that the UrbanGPT consistently retains its performance advantage, regardless of the LLM backbone employed. This confirms the efficacy of our spatio-temporal instruction-tuning paradigm, demonstrating its adaptability to various LLM backbones.

Table 4: Performance of UrbanGPT on the NYC-taxi and NYC-bike datasets using different LLMs as its backbone.

UrbanGPT	NYC-taxi				NYC-bike			
	Inflow		Outflow		Inflow		Outflow	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
w/ Llama	6.03	16.47	6.29	20.41	1.97	4.95	1.91	4.78
w/ Vicuna	6.16	16.92	6.83	21.78	2.02	5.16	2.01	5.03

A.4 Case Study

In this section, we assess the effectiveness of different large language models (LLMs) in zero-shot spatio-temporal prediction scenarios, as illustrated in Table 5 and Table 6. The instructions provided to the models are clearly indicated in blue font. The results demonstrate that various LLMs are capable of generating predictions based on these instructions, thereby highlighting the effectiveness of the prompt design. For instance, ChatGPT relies on historical averages rather than explicitly incorporating temporal or spatial data in its predictions. Llama-2-70b analyzes specific time periods and regions, but it encounters challenges in encoding numerical time-series dependencies, resulting in suboptimal predictive performance. On the other hand, Claude-2.1 effectively summarizes and analyzes historical data, leveraging peak-hour patterns and points of interest to achieve more accurate traffic trend predictions.

Our proposed UrbanGPT seamlessly integrates spatio-temporal contextual signals with the reasoning capabilities of large language models (LLMs) through a spatio-temporal instruction-tuning paradigm. This integration leads to remarkable improvements in predicting numerical values and spatio-temporal trends. These findings underscore the potential and effectiveness of our framework in capturing universal spatio-temporal patterns, making zero-shot spatio-temporal prediction practical and achievable.

Table 5: We examine the zero-shot predictions of different LLMs for bicycle flow in NYC with the provided instructions.

Instructions: Given the historical data for bike flow over 12 time steps in a specific region of New York City, the recorded bike inflows are [12 8 9 15 10 12 14 9 6 12 25 20], and the recorded bike outflows are [5 12 12 15 10 11 6 9 4 6 10 15]. The recording time of the historical data is 'January 14, 2020, 12:00, Tuesday to January 14, 2020, 17:30, Tuesday, with data points recorded at 30-minute intervals'. Here is the region information: This region is located within the Staten Island borough district and encompasses various POIs within a one-kilometer radius, covering Education Facility, Cultural Facility, Commercial, Transportation Facility categories. We now aim to predict the bike inflow and outflow for the next 12 time steps during the time period of 'January 14, 2020, 18:00, Tuesday to January 14, 2020, 23:30, Tuesday, with data points recorded at 30-minute intervals'. [To improve prediction accuracy, a spatio-temporal model is utilized to encode the historical bike data as tokens <ST_Start><ST_HIS><ST_HIS><ST_End>, where the first and the second tokens correspond to the representations of bike inflow and outflow.](#) Please conduct an analysis of the traffic patterns in this region, taking into account the provided time and regional information, and then generate the predictions ([the predictive tokens for regression](#)).

Ground true: Based on the given information, the predictions of bike inflow and outflow in this region are [26 38 28 24 20 13 16 9 5 3 5 2] and [12 19 10 10 11 9 6 5 4 4 2 2].

Llama-2-70b Responds: Based on the provided historical data, I have ... Here are my predictions for the bike inflow and outflow for the next 12 time steps: Bike Inflow: [15 18 20 15 12 10 14 12 10 15 18 20]. Bike Outflow: [10 12 15 10 11 12 10 9 6 10 12 15]. These predictions are based on the following analysis: Time of Day: The traffic patterns during this time are likely to be influenced by the evening rush hour, with increased bike inflows and outflows during the peak hours... Regional Information: ... This suggests that there may be a mix of both commuter and recreational bike traffic in the area...

ChatGPT Responds: To analyze the traffic patterns in the given region and make predictions for the next 12 time steps, we can calculate the average bike inflow and outflow based on the historical data... Now, we can generate predictions for the next 12 time steps: Predicted Bike Inflow: [11.67 11.67... 11.67]. Predicted Bike Outflow: [9.58 9.58... 9.58]...

Claude-2.1 Responds: Based on the provided historical data, this region in Staten Island seems to experience fluctuating yet relatively low bike traffic flow during the daytime on Tuesdays. The bike inflows range from 6 to 15 per 30-minute interval, while outflows range from 4 to 15. Given the region covers education, cultural, commercial, and transportation facilities within a 1km radius, the bike traffic likely corresponds to people traveling to and from these points of interest. The fluctuations could reflect class/work schedules and transportation needs throughout the day. For the evening prediction period of 6PM to 11:30PM, we may expect some further fluctuations but generally declining bike traffic as the day turns to night. People would be finishing up activities and heading home. I would predict the following bike inflows and outflows for the next 12 time steps: [10 8 7 6 5 4 3 2 1 1 1 1] and [8 6 5 4 3 3 2 2 1 1 1 1]...

UrbanGPT Responds: Based on the given information, the predictions ([predicted tokens](#)) of bike inflow and outflow in this region are <ST_PRE> [20 21 22 21 17 12 9 7 6 5 5] and <ST_PRE> [11 10 8 7 6 6 5 4 4 3 3 2].

Table 6: Zero-shot predictions of different LLMs for taxi flow in NYC with the provided instructions (without spatial information).

Instructions: Given the historical data for taxi flow over 12 time steps in a specific region of New York City, the recorded taxi inflows are [111 86 90 78 49 75 46 32 20 14 9 9], and the recorded taxi outflows are [100 91 58 39 47 47 19 22 21 9 16 10]. The recording time of the historical data is 'January 5, 2020, 20:30, Sunday to January 6, 2020, 02:00, Monday, with data points recorded at 30-minute intervals'. [No description is available for this region.](#) Now we want to predict the taxi inflow and outflow for the next 12 time steps during the time period of 'January 6, 2020, 02:30, Monday to January 6, 2020, 08:00, Monday, with data points recorded at 30-minute intervals'...

Ground true: Based on the given information, the predictions of taxi inflow and outflow in this region are [9 3 3 4 6 3 30 22 58 74 91 110] and [5 10 7 15 11 19 43 46 90 146 213 187].

Llama-2-70b Responds: Based on the provided historical data, I have analyzed the traffic patterns... Taxi Inflow: [105 90 80 70 60 55 45 35 25 15 10 5] Taxi Outflow: [95 85 70 60 50 45 35 25 15 10 5 0]. The predictions are based on the following factors...

ChatGPT Responds: To generate predictions for the next 12 time steps, we can use the average values as a baseline... Predicted taxi inflows: [53.33 53.33... 53.33]. Predicted taxi outflows: [40.83 40.83... 40.83]...

Claude-2.1 Responds: The data covers evening hours on a Sunday and early morning hours on a Monday. We can expect lower taxi demand during later Sunday evening, with some pickup around end-of-weekend times and early Monday commute times. The region is unspecified, so we cannot infer any location-specific usage patterns. Inflow ranges from 49-111 taxis over the 30 minute intervals. Outflow ranges from 19-100 taxis. This indicates the region likely serves both residential and commercial purposes. Prediction: Inflows: [25 15 10 8 7 9 15 25 35 40 45 50]. Outflows: [40 35 20 12 8 10 20 30 45 50 55 60]...

UrbanGPT Responds: Based on the given information, the predictions ([predicted tokens](#)) of taxi inflow and outflow in this region are <ST_PRE> [4 4 2 3 4 13 25 39 56 80 101 117] and <ST_PRE> [6 4 2 2 4 13 24 39 56 80 100 115].