UrbanMind: Urban Dynamics Prediction with Multifaceted Spatial-Temporal Large Language Models

Yuhang Liu
yliu41@binghamton.edu
State University of New York at Binghamton
Binghamton, NY, USA

Yingxue Zhang*
yzhang42@binghamton.edu
State University of New York at Binghamton
Binghamton, NY, USA

Xin Zhang
xzhang19@sdsu.edu
San Diego State University
San Diego, CA, USA

Ling Tian
lingtian@uestc.edu.cn
University of Electronic Science and Technology of China
Chengdu, China

Yanhua Li
yli15@wpi.edu
Worcester Polytechnic Institute
Worcester, MA, USA

Jun Luo
jluo@lscm.hk
Logistics and Supply Chain MultiTech R&D Centre
Hong Kong, China

Abstract

Understanding and predicting urban dynamics is crucial for managing transportation systems, optimizing urban planning, and enhancing public services. While neural network-based approaches have achieved success, they often rely on task-specific architectures and large volumes of data, limiting their ability to generalize across diverse urban scenarios. Meanwhile, Large Language Models (LLMs) offer strong reasoning and generalization capabilities, yet their application to spatial-temporal urban dynamics remains underexplored. Existing LLM-based methods struggle to effectively integrate multifaceted spatial-temporal data and fail to address distributional shifts between training and testing data, limiting their predictive reliability in real-world applications. To bridge this gap, we propose UrbanMind, a novel spatial-temporal LLM framework for multifaceted urban dynamics prediction that ensures both accurate forecasting and robust generalization. At its core, UrbanMind introduces Muffin-MAE, a multifaceted fusion masked autoencoder with specialized masking strategies that capture intricate spatial-temporal dependencies and intercorrelations among multifaceted urban dynamics. Additionally, we design a semantic-aware prompting and fine-tuning strategy that encodes spatial-temporal contextual details into prompts, enhancing LLMs' ability to reason over spatial-temporal patterns. To further improve generalization, we introduce a test time adaptation mechanism with a test data reconstructor, enabling UrbanMind to dynamically adjust to unseen test data by reconstructing LLM-generated embeddings. Extensive experiments on real-world urban dynamics datasets from multiple cities demonstrate the effectiveness of UrbanMind. The results consistently show that UrbanMind outperforms state-of-the-art baselines, achieving superior accuracy and strong generalization, even in zero-shot scenarios with no prior data.

*Corresponding author.

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737177

CCS Concepts

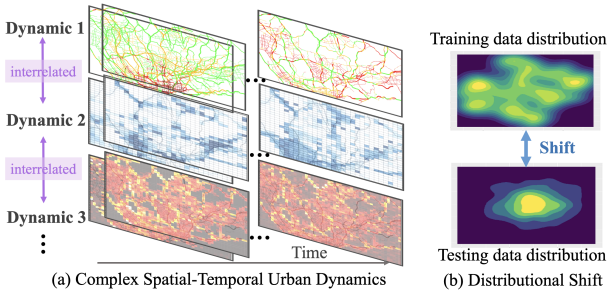
• **Computing methodologies** → **Learning latent representations**; **Temporal reasoning**.

Keywords

Urban Dynamics Prediction, Spatial-Temporal Data Mining, Large Language Models

ACM Reference Format:
Yuhang Liu, Yingxue Zhang, Xin Zhang, Ling Tian, Yanhua Li, and Jun Luo. 2025. 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 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737177

KDD Availability Link:
The source code of this paper has been made publicly available at https://doi.org/10.5281/zenodo.15484938.



(a) Complex Spatial-Temporal Urban Dynamics (b) Distributional Shift

Figure 1: Illustration of key challenges, including complex urban dynamics and distributional shifts.

1 Introduction

Urban dynamics prediction in traffic systems involves predicting human mobility patterns, such as traffic speed and travel demand, using historical spatial-temporal data collected from various IoT devices mounted on vehicles and public transports. Accurate prediction holds significant potential for optimizing traffic management, enhancing urban planning, and improving public service delivery.

Limitations of NN-based approaches. Various deep neural network (NN)-based approaches have been developed for urban dynamics prediction. Graph-based neural networks [11, 45, 59] have demonstrated exceptional success in modeling spatial-temporal dependencies. Generative models, including GAN-based methods [51, 53, 55] and diffusion-based models [10, 27, 34], have gained significant attention for their ability to model complex urban data distributions. Meta-learning and transfer-learning-based approaches [43, 50, 54, 56, 57] have also been explored for their flexibility and adaptability. Despite these advances, existing methods often rely heavily on task-specific architectures and substantial volumes of data. These dependencies limit their ability to generalize effectively across diverse urban contexts or adapt to evolving conditions, particularly in scenarios involving unseen regions or sparse data. This limitation raises the question: *How can we enhance urban dynamics prediction by simultaneously achieving high accuracy and strong generalization across dynamic and diverse urban scenarios?*

Limitations of LLM-based approaches. Considering the outstanding generalization ability, we turn attention to Large Language Models (LLMs) [28]. LLMs have revolutionized natural language processing with their exceptional reasoning, prediction, and generalization capabilities, enabling adaptability across diverse applications with minimal task-specific training. Despite their success in other fields, the application of LLMs in urban dynamics prediction from spatial-temporal data remains relatively unexplored, with only a few notable attempts. We term these attempts as LLM-based approaches. For instance, UrbanGPT [19] integrates a spatial-temporal dependency encoder with an instruction-tuning paradigm in LLMs. However, it does not consider the inter-correlations among different urban dynamics, limiting its predictive performance in complex, multifaceted urban scenarios. Similarly, methods such as ST-LLM [23], TPLLM [32], GATGPT [3], and STG-LLM [25] attempt to adapt LLMs to spatial-temporal data. However, these models lack the capability to utilize prompts for prediction guidance and fail to address distributional shifts between training and testing data, reducing their flexibility and adaptability in diverse urban settings.

Our objectives and challenges. Building on the strengths and limitations of NN- and LLM-based approaches, this paper proposes a novel method for adapting LLMs to the spatial-temporal domain for multifaceted urban dynamics prediction, achieving high accuracy while demonstrating strong generalization across diverse urban scenarios, including those with no prior data or unseen conditions. Achieving this goal requires overcoming below challenges:

Challenge 1: Bridging Spatial-Temporal Data and LLMs: As shown in Figure 1(a), spatial-temporal urban data are inherently complex, exhibiting structured dependencies across space, time, and multiple dynamic factors. However, LLMs are not inherently designed to process such structured, continuous signals, especially time-series data [39]. This creates a fundamental representation gap: raw spatial-temporal data must be transformed into discrete, semantically meaningful token representations to be interpretable by LLMs. Bridging this gap is crucial for enabling LLMs to reason over urban dynamics.

Challenge 2: Prompt Design and Fine-Tuning: Spatial and temporal information inherently contain rich semantic details that enhance a model's understanding of spatial-temporal patterns. Effectively leveraging these details to design prompts for LLM fine-tuning is crucial for enabling LLMs to comprehend spatial-temporal dynamics within specific urban contexts, facilitating reasoning over the data and producing reliable forecasts.

Challenge 3: Distributional Shift: Urban dynamics often exhibit substantial distributional shifts between training and testing data as shown in Figure 1(b), particularly in zero-shot scenarios where the model is tested in unseen settings with no prior data available. These shifts, often reflecting significant differences in geographical characteristics, or temporal patterns compared to the training data, pose a major barrier to generalization and can severely degrade prediction performance.

Our UrbanMind. To address these challenges, we propose UrbanMind, a novel multifaceted spatial-temporal LLM that can achieve high prediction accuracy and robust generalization in diverse urban scenarios. UrbanMind integrates a Mask-Empowered Representation Learning framework through the novel Muffin-MAE, which employs specialized masking mechanisms to capture complex spatial-temporal dependencies and intercorrelations among multifaceted urban dynamics. Additionally, it incorporates a tailored prompting and fine-tuning strategy for spatial-temporal data. The prompt encodes spatial-temporal semantic information, guiding the fine-tuning process of the LLM. To generate predictions, a predictor module is attached to the LLM, transforming the high-dimensional latent vectors generated by the LLM into spatial-temporal urban dynamics forecasts. To address distributional shifts, UrbanMind features a novel test time adaptation strategy with a data reconstructor that operates parallel to the predictor and shares layers with it. During testing, this module adapts to test data by reconstructing the LLM output and fine-tuning the shared layers. This ensures improved alignment with the test data distribution, thereby enhancing the generalization capability of the predictor. *The primary contributions of this paper can be summarized as follows:*

- We propose a novel multifaceted spatial-temporal LLM, UrbanMind, which integrates innovative designs to effectively enable a language model to proficiently understand and process the intricate relationships and patterns inherent in multifaceted spatial-temporal data. UrbanMind achieves high prediction accuracy and robust generalization in diverse urban scenarios, including zero-shot cases with no prior data available.
- UrbanMind incorporates a novel multifaceted fusion masked autoencoder (Muffin-MAE) with advanced masking strategies to generate embeddings that capture both spatial-temporal dependencies and multifaceted intercorrelations, enabling seamless integration with LLMs. Additionally, a semantic-rich prompt and fine-tuning strategy tailored for spatial-temporal data is designed, along with an innovative test time adaptation mechanism that mitigates distributional shifts through a test data reconstructor.
- We conducted extensive experiments on three different urban dynamics—traffic speed, inflow, and travel demand—in three different cities. The results demonstrate the proposed UrbanMind's exceptional ability to generalize across diverse spatial-temporal learning scenarios and deliver accurate predictions under various conditions, outperforming state-of-the-art baselines, even in

unseen regions or scenarios with no prior training data. *We have also made our data and code available to the research community* [1].

2 Problem Definition

Definition 1 (Grid Cell s_{ij}). To achieve a more granular understanding of urban dynamics, a city is partitioned into defined grid cells. For instance, the city can be divided into $J \times J$ grid cells, each with uniform dimensions (*e.g.*, $1 \times 1, \text{km}^2$). The set of all grid cells is represented as $\mathcal{S} = \{s_{ij}\}$, where $1 \le i \le J$ and $1 \le j \le J$.

Definition 2 (Target Region r_{ij}). Each grid cell s_{ij} corresponds to a target region r_{ij}. A target region r_{ij} is a square geographic area consisting of $\ell \times \ell$ grid cells, with the cell s_{ij} positioned at its top-left corner. Formally, a target region is represented as $r_{ij} = \langle s_{ij}, \ell \rangle$. The set of all regions is denoted as $\mathcal{R} = \{r_{ij}\}_{1 \le i,j \le J}$.

Definition 3 (Urban Dynamics $\mathcal{X}$ and $\mathcal{X}^k$). Urban dynamics of a region r encompass various aspects, such as traffic speed, vehicle inflow and outflow, and human mobility, among others. We represent the urban dynamics for a region r as a five-dimensional tensor $\mathcal{X} \in \mathbb{R}^{N \times T \times C \times \ell \times \ell}$, where N is the number of days, T represents the number of time steps per day (e.g., hours), C denotes the number of urban dynamic aspects, referred to as the number of channels, and $\ell \times \ell$ specifies the spatial dimensions of the region. We further denote the target urban dynamics of interest for a region r as $\mathcal{X}^k$, where $\mathcal{X}^k \in \mathbb{R}^{N \times T \times 1 \times \ell \times \ell}$, representing a single-channel subset of $\mathcal{X}$ corresponding to the specific dynamic being analyzed.

Problem Definition. Given the historical, multifaceted urban dynamics data $\mathcal{X}^{\text{tr}}$ for training regions denoted as $\mathcal{R}^{\text{tr}} = \{r_{ij}^{\text{tr}}\}$, the goal is to train a generalizable function f^* that is able to adapt to testing regions $\mathcal{R}^{\text{ts}} = \{r_{ij}^{\text{ts}}\}$. Specifically, the function f^* is able to predict the urban dynamics for the next m hours in testing regions $\mathcal{R}^{\text{ts}} = \{r_{ij}^{\text{ts}}\}$, using h hours of prior observations $\mathcal{X}^{\text{ts}}_{\text{start}}$ from these testing regions. Formally, the task is expressed as:

$$\hat{\mathcal{X}}^{\text{ts}} = f^*\left(\mathcal{X}^{\text{ts}}_{\text{start}}\right), \text{where } f^* = \arg\min_f \mathcal{L}(f, \mathcal{X}^{\text{tr}}),$$

where $\hat{\mathcal{X}}^{\text{ts}}$ denotes the predicted urban dynamics for the following m hours, and $\mathcal{L}$ is the loss function while training the f function. *Note:* In the standard spatial-temporal prediction scenario, $\mathcal{R}^{\text{ts}} \subseteq \mathcal{R}^{\text{tr}}$. In the zero-shot prediction scenario, the testing regions $\mathcal{R}^{\text{ts}}$ must be unseen during the training process and completely distinct from $\mathcal{R}^{\text{tr}}$, *i.e.*, $\mathcal{R}^{\text{ts}} \cap \mathcal{R}^{\text{tr}} = \emptyset$.

3 Methodology

In this section, we introduce UrbanMind, a novel spatial-temporal LLM designed to advance multifaceted urban dynamics prediction performance and enhance adaptability and generalization. It comprises three components, each addressing the challenges outlined in Section 1: (1) *Mask-Empowered Representation Learning.* UrbanMind includes a novel Muffin-MAE with specifically designed masking mechanisms to project urban dynamics into a latent space, effectively capturing complex spatial-temporal dependencies and interdependencies among multifaceted urban dynamics (see Section 3.1). (2) *Semantic-Aware Prompting and Fine-Tuning.* UrbanMind designs semantic-aware prompts that encode detailed spatial-temporal contexts. These prompts guide the fine-tuning of the LLM, enabling it to reason over the spatial-temporal embeddings (see Section 3.2). (3) *Test time adaptation.* UrbanMind incorporates a data reconstructor that quickly adapts to test data by reconstructing the LLM's output. This enables the model to better align with the test data distribution, enhancing its generalization capability (see Section 3.3).

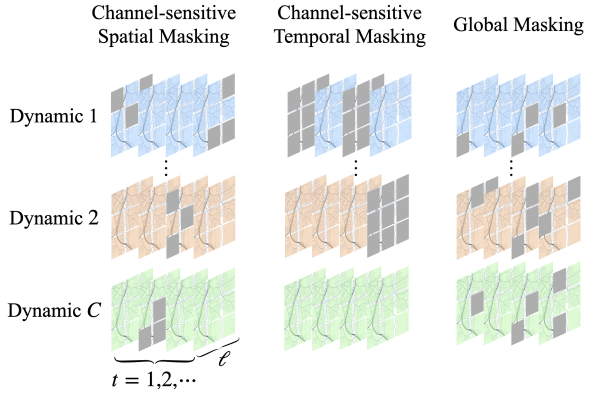


Figure 2: Masking strategies.

3.1 Mask-Empowered Representation Learning

Mask-Empowered Representation Learning is a critical component of UrbanMind. While LLMs are inherently designed to process natural language data, such as text in the form of tokens, they are not equipped to directly handle spatial-temporal data. To leverage LLMs for urban dynamics prediction, it is crucial to transform spatial-temporal urban dynamics into latent representations that LLMs can effectively process and understand, while preserving both the spatial-temporal dependencies and the intercorrelations among multifaceted urban dynamics.

Multifaceted Fusion MAE (Muffin-MAE). To achieve this, we propose a multifaceted fusion masked autoencoder (Muffin-MAE), a framework with specifically designed masking mechanisms. The Muffin-MAE architecture, illustrated in Figure 3, consists of two masked autoencoders. The first, comprising an encoder E_1 and a decoder D_1, generates multifaceted embeddings by processing multifaceted urban dynamics, effectively capturing their interdependencies. The second, with a separate encoder E_2 and decoder D_2, generates target embeddings for individual urban dynamics, preserving the spatial-temporal dependencies of the target dynamic.

Both encoders and decoders consist of multiple convolutional layers, enabling effective processing across spatial and temporal dimensions. The encoder E_1 takes as input a sequence of multifaceted masked urban dynamics data, $\mathcal{X}^{\text{masked}} = \{\boldsymbol{X}_t^{\text{masked}}\}_{t=1}^T$, where specialized masking mechanisms selectively mask portions of the data. It outputs a sequence of multifaceted latent embeddings, *i.e.*, $\mathcal{V} = E_1\left(\mathcal{X}^{\text{masked}}\right)$, where $\mathcal{V} = \{\boldsymbol{v}_t\}_{t=1}^T$ with one embedding generated per time slot. These embeddings are then passed to the decoder D_1, which reconstructs the original multifaceted urban dynamics data, $\hat{\mathcal{X}} = D_1(\mathcal{V}) = \{\hat{\boldsymbol{X}}_t\}_{t=1}^T$. The objective follows:

$$\mathcal{L}_{\text{Muffin-MAE}} = \frac{1}{T}\sum_{t=1}^{T}\left(\hat{\boldsymbol{X}}_t - \boldsymbol{X}_t\right)^2. \quad (1)$$

[1] UrbanMind code: https://doi.org/10.5281/zenodo.15484938.

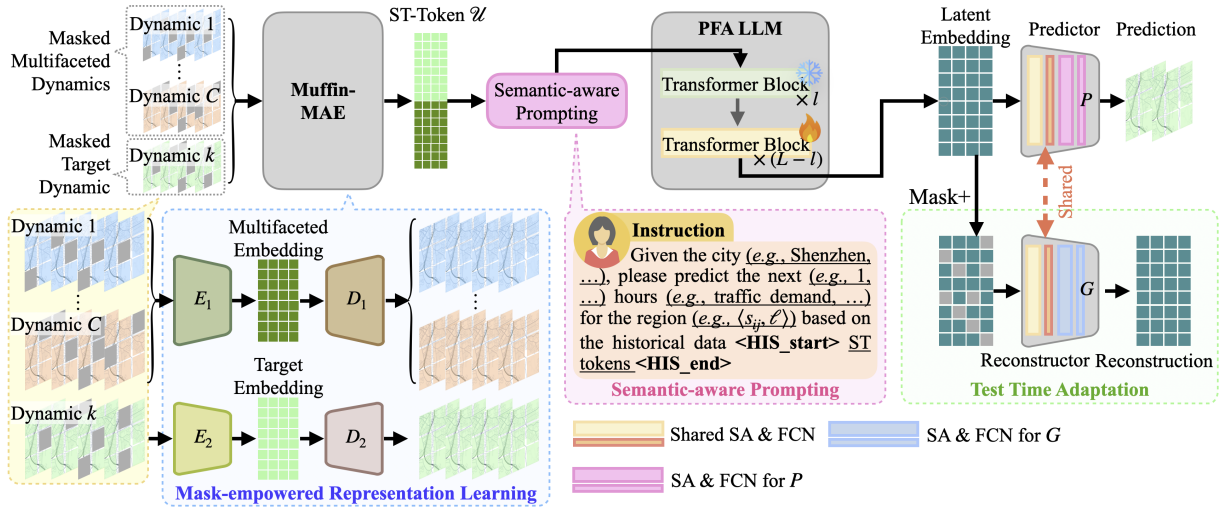


Figure 3: UrbanMind Framework.

Once E_1 and D_1 are trained on multifaceted urban dynamics, as shown in Figure 3, E_2 and D_2 operate similarly but process sequences of individual target urban dynamics, $\mathcal{X}^k \in \mathbb{R}^{N\times T\times 1\times \ell\times \ell}$, where the number of channels is set to 1, reflecting the separation of dynamics.

Unlike the masking mechanisms in standard MAE [9], which primarily rely on space-only or time-only random sampling broadcasted across dimensions, our Muffin-MAE introduces specialized masking strategies designed to capture both spatial and temporal dependencies across different urban dynamics, as illustrated in Figure 2 and detailed below.

- **Channel-Sensitive Spatial Masking:** For a sequence of urban dynamics $\mathcal{X} = \{X_{r,t}\}_{t=1}^{T}$ at a region r with $X_{r,t} \in \mathbb{R}^{C\times l\times l}$, a channel $c \in \{1,\dots,C\}$ is randomly selected for each time step t. Within the selected channel $X_{r,t}^{c} \in \mathbb{R}^{l\times l}$, p_s of the $\ell\times\ell$ dynamics features are randomly masked with $p_s \in (0,1)$. Let $\mathcal{M}_{\text{spatial}}$ denote the set of masked feature indices, and $|\mathcal{M}_{\text{spatial}}| = p_s \cdot l^2$. The masked tensor is given by:

$$X_{r,t}^{c,\text{masked}}(i,j) = \begin{cases} 0, & \text{if } (i,j) \in \mathcal{M}_{\text{spatial}}, \\ X_{r,t}^{c}(i,j), & \text{otherwise.} \end{cases}$$

  This mechanism ensures that different grid cells of a region are masked across channels, enabling the Muffin-MAE to effectively capture spatial dependencies.
- **Channel-Sensitive Temporal Masking:** For the sequence $\mathcal{X} = \{X_{r,t}\}_{t=1}^{T}$, p_t of time steps are randomly sampled from T with $p_t \in (0,1)$. For each sampled time step t^{m}, a channel $c \in \{1,\dots,C\}$ is randomly selected, and all information corresponding this channel at this region is masked. The masking process for the selected channel is:

$$X_{r,t^{\text{m}}}^{c,\text{masked}}(i,j) = 0, \quad \forall (i,j) \in \{1,\dots,l\}\times\{1,\dots,l\}.$$

  This approach allows the Muffin-MAE to flexibly mask different time steps in different channels, facilitating the capture of temporal dependencies among multi-faceted urban dynamics.
- **Global Masking:** For the sequence $\mathcal{X} = \{X_{r,t}\}_{t=1}^{T}$, p_t of time steps are randomly sampled. For each sampled time step t^{sp}, p_s of grid cells across all channels are randomly masked. Let $\mathcal{M}_{\text{global}}$ denote the set of masked grid cell indices, where $|\mathcal{M}_{\text{global}}| = p_s \cdot C \cdot l^2$. The masked data is given by:

$$X_{r,t}^{\text{masked}}(c,i,j) = \begin{cases} 0, & \text{if } (c,i,j) \in \mathcal{M}_{\text{global}}, \\ X_{r,t}(c,i,j), & \text{otherwise.} \end{cases}$$

  Global masking mechanism enhances Muffin-MAE to capture both spatial and temporal dependencies across urban dynamics.

Multifaceted and Target Embeddings. To obtain the multifaceted embeddings from the multifaceted urban dynamics $\mathcal{X}$, we train the encoder E_1 and decoder D_1 in Muffin-MAE using all three masking mechanisms. Upon completion of training, the encoder generates embeddings $\mathcal{V}$, effectively preserving the inter-correlations among the related multi-faceted urban dynamics. To derive target embeddings $\mathcal{V}^k = \{v_{r,t}^k\}_{t=1}^{T}$ for a sequence of target urban dynamics $\mathcal{X}^k$, we train the encoder E_2 and the decoder D_2 using the target urban dynamics. During this training process, all three masking mechanisms are again applied to the target urban dynamics, with the channel number set to 1.

Spatial-Temporal Token Generation. To form the final tokens for a sequence of urban dynamics, we combine the multifaceted embeddings $\mathcal{V} = \{v_{r,t}\}_{t=1}^{T}$ with the target embeddings $\mathcal{V}^k = \{v_{r,t}^k\}_{t=1}^{T}$. This combination is achieved by concatenating the embeddings along the feature dimension for each time step as $\mathcal{U} = \{u_{r,t}\}_{t=1}^{T}$, where $u_{r,t} = \text{concat}(v_{r,t}, v_{r,t}^k)$. Here, $u_{r,t}$ represents the final token for time step t, encapsulating both the inter-correlations among the related multifacet dynamics and the spatial-temporal dependencies of the target urban dynamics. These tokens, structured as $u_{r,t} \in \mathbb{R}^{d_v+d_k}$, are well-suited for processing by LLMs.

3.2 Semantic-Aware Prompting and Fine-Tuning

Semantic-Aware Prompting Design. In spatial-temporal prediction tasks, both temporal and spatial information carry essential semantic details that contribute to the model's understanding of complex patterns within specific contexts. To leverage these insights, we represent temporal and spatial information as prompt

instruction text, enabling large language models to process this data effectively through their advanced text comprehension capabilities.

In the UrbanMind framework, we incorporate time, regional, and task-specific information into the instruction input for the large language model. Temporal information includes data from the previous hours of a day and the target hours whose urban dynamics need to be predicted. Regional information specifies the city, the coordinates of the top-left grid cell s_{ij} within the target region, and the region's side length ℓ. Additionally, the task name defines the specific urban dynamics to be predicted. By encapsulating these elements in a structured prompt, UrbanMind effectively identifies and assimilates spatial-temporal patterns across diverse regions, time frames, and urban dynamics as is illustrated in Figure 3.

LLM Fine-Tuning Strategy. To adapt the LLM for spatial-temporal urban dynamics prediction, we employ the Partially Frozen Attention (PFA) mechanism [23]. Consider an LLM (we use LLaMA3.2 [8] in this paper) with L transformer layers, where each transformer layer TFM_i $(i = 1, \ldots, L)$ undergoes the following transformation:

$$\boldsymbol{e}^{(i)} = \text{LayerNorm}(\boldsymbol{e}^{(i-1)} + \text{SA}(\boldsymbol{e}^{(i-1)})),$$

where $\boldsymbol{e}^{(i)}$ represents the hidden state produced by the i-th layer of the LLM, and $\text{SA}(\cdot)$ denotes the self-attention operation.

To efficiently fine-tune the model, we divide the transformer layers into two groups: frozen layers $\text{TFM}^{\text{fr}} = \{\text{TFM}^{(1)}, \ldots, \text{TFM}^{(l)}\}$ and trainable layers $\text{TFM}^{\text{tr}} = \{\text{TFM}^{(l+1)}, \ldots, \text{TFM}^{(L)}\}$. The parameters of the frozen layers TFM^{fr} remain fixed during fine-tuning, preserving the pretrained knowledge of the LLM. The trainable layers TFM^{tr} process the output from the frozen layers, where only the query matrices $\mathbf{W}_q$ in the self-attention mechanism are updated to learn task-specific spatial-temporal dependencies. In contrast, the key and value matrices $\mathbf{W}_k$ and $\mathbf{W}_v$ remain frozen to retain the pretrained relationships encoded in the model. This two-stage fine-tuning process leverages the generalization capabilities of the frozen layers while allowing the trainable layers to effectively adapt to the spatial-temporal characteristics of urban dynamics.

Spatial-Temporal Predictor Module. The LLM's output—a high-dimensional sequence of latent embeddings—must be transformed into numerical values representing predicted urban dynamics. To realize this goal, we attach a spatial-temporal predictor module P to the LLM. This module consists of self-attention layers and fully connected layers (in Figure 3), enabling the transformation of LLM-generated embeddings into structured urban dynamics predictions.

The spatial-temporal prediction module P processes the sequence of embeddings $\mathcal{E} = \{\boldsymbol{e}\}$ generated by the LLM based on the prompt using h hours of prior observations and predicts the urban dynamics $\hat{\mathcal{X}}^k = \{\hat{\boldsymbol{X}}^k_{r,t}\}_{t=h+1}^{h+m}$ for the subsequent m hours, where $\hat{\mathcal{X}}^k = P(\mathcal{E})$. To optimize the prediction, we minimize the Mean Squared Error (MSE) loss for daily sequential urban dynamics data:

$$\mathcal{L}_{\text{pred}} = \frac{1}{m} \sum_{t=h+1}^{h+m} \left\| \tilde{\boldsymbol{X}}^k_{r,t} - \boldsymbol{X}^k_{r,t} \right\|^2, \tag{2}$$

where $\hat{\boldsymbol{X}}^k_{r,t}$ represents the predicted urban dynamics, and $\boldsymbol{X}^k_{r,t}$ denotes the corresponding ground truth values.

3.3 Test Time Adaptation

Although LLMs exhibit strong generalizability, they are primarily designed for natural language tasks and generalization across text-related domains. In the spatial-temporal domain, testing data often presents distinct patterns from training data, especially in zero-shot scenarios where unseen regions and varying traffic dynamics introduce significant distributional shifts. To address this, we propose a test time adaptation strategy to enhance the generalization and adaptability of LLMs during testing. At its core is a novel test data reconstructor G, which works in parallel with the predictor while sharing several self-attention layers as is shown in Figure 3.

During testing, the LLM processes the prompt describing the test region (potentially unseen) and generates a latent embedding sequence $\mathcal{E} = \{\boldsymbol{e}\}$ as input to the predictor. Before directly passing $\mathcal{E}$ to the predictor, we randomly mask p of the elements (with $p \in (0, 1)$ as the masking ratio) in $\mathcal{E}$ to introduce stochasticity and encourage robustness. The reconstructor G then recovers the masked elements through a quick adaptation process, performing a few epochs of updates. The masking process works as follows: A binary mask vector $\boldsymbol{m}_i \in \{0, 1\}^d$ is generated for each $\boldsymbol{e}_i \in \mathbb{R}^d$. The indices to be masked, $\mathcal{M} \subseteq \{1, \ldots, d\}$, are randomly sampled with a uniform distribution such that $|\mathcal{M}| = p \cdot d$.], *i.e.*,

$$\boldsymbol{e}_i^{\text{masked}} = \boldsymbol{e}_i \odot \boldsymbol{m}_i, \tag{3}$$

where $\odot$ denotes element-wise multiplication. The sequence of masked embeddings is then represented as $\mathcal{E}^{\text{masked}} = \{\boldsymbol{e}_i^{\text{masked}}\}$. The reconstructor G reconstructs the entire sequence of latent embeddings $\mathcal{E} = \{\boldsymbol{e}\}$ from the masked sequence $\mathcal{E}^{\text{masked}}$ with loss:

$$\mathcal{L}_{\text{recon}} = \frac{1}{n} \sum_{i=1}^{n} \left\| G(\boldsymbol{e}_i^{\text{masked}}) - \boldsymbol{e}_i \right\|^2, \tag{4}$$

where n is the number of embeddings, $\boldsymbol{e}_i$ is the original embedding, and $G(\boldsymbol{e}_i^{\text{masked}})$ is the reconstructed embedding produced by the reconstructor G. This reconstruction process allows G to quickly adapt to new data and regional conditions, fine-tuning the shared layers to better align with the test data. Once adaptation is complete, the updated shared layers enable the predictor to generate more accurate results for testing scenarios. This approach effectively mitigates distributional shifts, enhancing both generalization and adaptability in diverse spatial-temporal contexts[37, 38]. The detailed algorithm for UrbanMind can be found in Algorithm 1.

4 Experiment

In experiments, we aim to measure the effectiveness of our UrbanMind across different datasets and in different urban scenarios. We will answer the following questions with extensive experiments: (1) Can our UrbanMind outperform in zero-shot spatial-temporal prediction tasks for different urban dynamics? (2) Can our UrbanMind outperform in standard spatial-temporal prediction tasks for different urban dynamics? (3) Is each component in UrbanMind effective when performing spatial-temporal predictions? (4) How do different hyperparameters affect performance?

4.1 Dataset and Experiment Descriptions

Data description. In our experiments, we use three distinct urban dynamics datasets—travel demand, traffic speed, and inflow—from

City	Dynamics	Metrics	Models											
			DYffusion	TGC-LSTM	GCRN	GAGCN	GATGPT	GCNGPT	ST-LLM	TPLLM	LLaMA3	STG-LLM	UrbanGPT	UrbanMind
Shenzhen	Speed	MAE	**0.131**	0.196	0.196	0.361	0.139	0.159	0.194	0.166	0.256	0.194	0.132	**0.131**
		RMSE	0.211	0.263	0.265	0.485	0.205	0.225	0.290	0.229	0.443	0.274	0.201	**0.194**
	Inflow	MAE	0.128	0.235	0.324	0.371	0.128	0.137	0.153	0.165	0.269	0.217	0.137	**0.127**
		RMSE	0.251	0.364	0.489	0.542	0.250	0.254	0.265	0.255	0.460	0.382	**0.249**	**0.249**
	Demand	MAE	0.149	0.228	0.324	0.464	0.141	0.153	0.207	0.178	0.223	0.202	0.144	**0.138**
		RMSE	0.339	0.378	0.504	0.644	0.270	0.275	0.301	0.278	0.381	0.377	0.271	**0.269**
Xi'an	Speed	MAE	0.216	0.201	0.230	0.481	0.268	0.138	0.220	0.198	0.294	0.208	0.139	**0.137**
		RMSE	0.253	0.227	0.263	0.314	0.296	0.183	0.323	0.265	0.417	0.228	0.192	**0.162**
	Inflow	MAE	0.386	0.263	0.419	0.360	0.300	0.142	0.231	0.330	0.356	0.334	0.170	**0.114**
		RMSE	0.552	0.392	0.452	0.422	0.341	0.194	0.340	0.460	0.530	0.414	0.237	**0.173**
	Demand	MAE	0.330	0.291	0.291	0.303	0.337	0.210	0.195	0.333	0.279	0.263	0.185	**0.160**
		RMSE	0.314	0.430	0.358	0.416	0.384	0.341	0.290	0.441	0.435	0.285	0.284	**0.273**
Chengdu	Speed	MAE	0.201	0.157	0.202	0.460	0.253	0.135	0.302	0.185	0.289	0.175	0.133	**0.120**
		RMSE	0.240	0.204	0.251	0.501	0.282	0.190	0.349	0.238	0.463	0.194	0.191	**0.166**
	Inflow	MAE	0.309	0.328	0.550	0.247	0.334	0.191	0.217	0.348	0.349	0.308	0.203	**0.187**
		RMSE	0.520	0.454	0.646	0.361	0.379	0.243	0.324	0.469	0.480	0.389	0.262	**0.240**
	Demand	MAE	0.376	0.284	0.452	0.297	0.259	0.171	0.169	0.290	0.264	0.356	0.160	**0.125**
		RMSE	0.406	0.405	0.486	0.416	0.301	0.230	0.233	0.388	0.481	0.381	0.216	**0.202**

Table 1: Zero-shot Prediction: Average performance for speed, inflow, and demand predictions across 3 cities. In this zero-shot case, all testing regions are unseen during training.

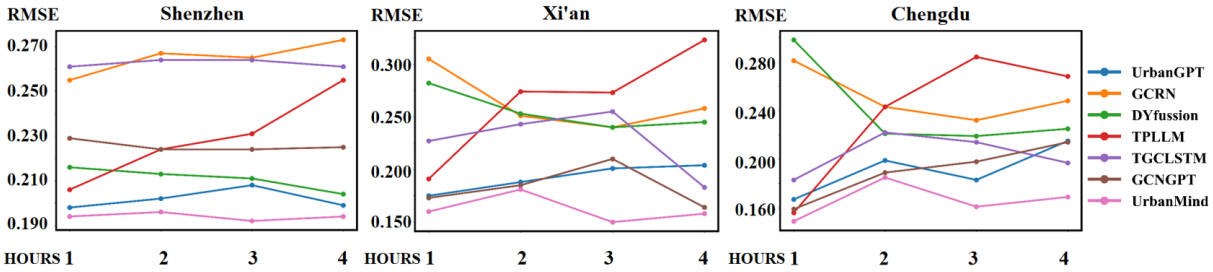


Figure 4: Zero-shot Prediction: RMSE for traffic speed prediction over 4 hours in Shenzhen, Xi'an, and Chengdu.

multiple cities, including Chengdu, Xi'an, and Shenzhen. For each city, the entire area is partitioned into equal-sized grid cells with a region size of 10×10. In Shenzhen, each dataset is processed to yield dimensions of $(162, 12, 63, 10, 10)$, where 162 represents the number of days, 12 corresponds to the twelve one-hour time slots per day, and 63 indicates the number of regions. For Chengdu and Xi'an, each type of urban dynamics is structured with dimensions $(30, 12, 4, 10, 10)$, where 30 denotes the number of days, 12 indicates the time slots per day and 4 represents the number of regions. All the data and regions in all datasets are divided into training and testing. More details can be found in the Appendix.

Task Description. To validate the prediction accuracy and generalizability of UrbanMind, we conduct experiments on two tasks: *(1) Zero-shot Prediction.* We evaluate the model's generalizability by predicting future spatial-temporal data from unseen regions in three cities for all three urban dynamics types. *(2) Standard Prediction.* We assess the model's prediction accuracy by predicting future urban dynamics for the same regions included in the training set.

4.2 Baselines

We compare UrbanMind against state-of-the-art urban dynamics prediction models, including LLM-based approaches: **GATGPT** [4], **GCNGPT** [4, 24], **ST-LLM** [24], **TPLLM** [33], **UrbanGPT** [20], and **STG-LLM** [26]. These models integrate LLMs with CNNs or GCNs to enhance urban dynamics prediction. Additionally, we use **LLaMA3.2** [8] as UrbanMind's foundation model, referred to as **LLaMA3**. It is also evaluated as a baseline to show the limits of directly applying LLMs to urban dynamics prediction. Beyond these, we also benchmark against most recent and effective neural network-based models, including **DYffusion** [34], **TGC-LSTM** [6], **GCRN** [35], and **GAGCN** [42]. More details on the baselines and experimental settings are provided in the Appendix.

Evaluation Metrics: All models are evaluated using MAE and RMSE to assess urban dynamics prediction performance.

4.3 Experimental Settings

For Mask-Empowered Representation Learning and Semantic-Aware Prompting and Fine-Tuning stages, we employed the Adam optimizer with a learning rate of 0.00001 for the Shenzhen dataset and 0.0001 for the Xi'an and Chengdu datasets. In the Test Time Adaptation stage, the Adam optimizer was applied with a learning rate of 0.00005 for Shenzhen and 0.0005 for Xi'an and Chengdu. Additionally, LLaMA3 serves as the foundation for UrbanMind.

City	Dynamics	Metrics	Models											
			DYffusion	TGC-LSTM	GCRN	GAGCN	GATGPT	GCNGPT	ST-LLM	TPLLM	LLaMA3	STG-LLM	UrbanGPT	UrbanMind
Shenzhen	Speed	MAE	0.175	0.187	0.202	0.307	0.155	0.161	0.201	0.225	0.261	0.167	0.147	**0.130**
		RMSE	0.239	0.218	0.235	0.515	0.203	0.230	0.281	0.248	0.454	0.245	0.205	**0.200**
	Inflow	MAE	0.272	0.142	0.132	0.182	0.136	0.301	0.286	0.181	0.268	0.240	0.128	**0.123**
		RMSE	0.443	0.207	0.189	0.271	0.235	0.389	0.381	0.211	0.450	0.400	0.190	**0.185**
	Demand	MAE	0.285	0.166	0.168	0.338	0.154	0.228	0.223	0.178	0.236	0.371	0.137	**0.133**
		RMSE	0.389	0.257	0.220	0.518	0.275	0.343	0.325	0.222	0.380	0.464	0.210	**0.199**
Xi'an	Speed	MAE	0.210	0.156	0.225	0.421	0.304	0.291	0.425	0.270	0.317	0.183	0.134	**0.114**
		RMSE	0.266	0.201	0.264	0.524	0.358	0.415	0.520	0.348	0.516	0.220	0.190	**0.186**
	Inflow	MAE	0.304	0.254	0.430	0.341	0.311	0.298	0.289	0.334	0.363	0.350	0.214	**0.124**
		RMSE	0.533	0.384	0.512	0.546	0.367	0.437	0.368	0.442	0.551	0.383	0.330	**0.181**
	Demand	MAE	0.259	0.329	0.378	0.377	0.381	0.401	0.199	0.382	0.327	0.359	0.253	**0.159**
		RMSE	0.439	0.466	0.495	0.501	0.431	0.525	0.281	0.489	0.513	0.401	0.407	**0.273**
Chengdu	Speed	MAE	0.193	0.137	0.216	0.516	0.289	0.292	0.448	0.229	0.293	0.152	0.147	**0.119**
		RMSE	0.244	0.182	0.255	0.627	0.331	0.396	0.528	0.289	0.480	0.184	0.170	**0.165**
	Inflow	MAE	0.361	0.287	0.315	0.320	0.234	0.251	0.326	0.285	0.343	0.397	0.267	**0.153**
		RMSE	0.521	0.381	0.451	0.433	0.287	0.338	0.409	0.387	0.490	0.478	0.389	**0.191**
	Demand	MAE	0.306	0.271	0.304	0.378	0.251	0.194	0.236	0.257	0.289	0.316	0.229	**0.153**
		RMSE	0.491	0.390	0.402	0.493	0.392	0.236	0.321	0.358	0.479	0.375	0.340	**0.187**

Table 2: Standard Prediction: Average performance of speed, inflow, and demand prediction across 3 cities. Historical data for testing regions were included during training, and future dynamics are predicted.

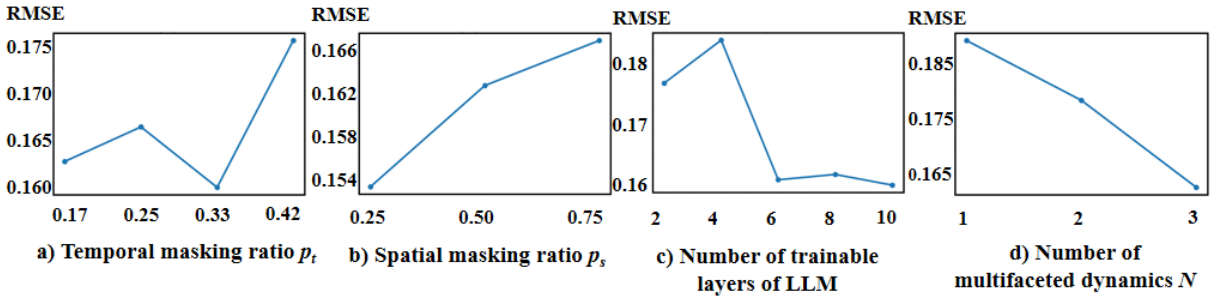


Figure 5: Impact of Hyperparameters on RMSE Performance for Traffic Speed Prediction in Xi'an.

4.4 Empirical Results

Results of Question (1) (Zero-shot Performance). To evaluate UrbanMind's performance in zero-shot dynamics prediction, we conducted experiments on three urban dynamics datasets across Shenzhen, Xi'an, and Chengdu. Tab. 1 presents the average prediction results, while Fig. 4 illustrates the detailed prediction performance over time. UrbanMind consistently outperforms all baselines across datasets and metrics, achieving the lowest MAE and RMSE values, demonstrating outstanding generalization capability. In Shenzhen's speed prediction, while DYffusion achieved a comparable MAE to UrbanMind, its higher RMSE suggests greater prediction variance. Similarly, for Shenzhen's inflow prediction, UrbanGPT matched UrbanMind in RMSE but underperformed in MAE, highlighting UrbanMind's superior accuracy. UrbanMind's strength lies in its ability to generalize across diverse traffic forecasting scenarios, leveraging LLMs' generalization capability further enhanced by the test time adaptation mechanism, ensuring exceptional robustness in zero-shot urban traffic prediction tasks.

Results of Question (2) (Standard Prediction Performance). To evaluate UrbanMind's performance in standard spatial-temporal prediction tasks, we present the results for multiple urban dynamics across cities in Tab. 2. UrbanMind consistently outperforms baselines, achieving the lowest MAE and RMSE across all datasets and metrics. In Shenzhen's traffic speed prediction, UrbanMind demonstrated superior accuracy and consistency compared to DYffusion, with lower MAE and RMSE values. Similarly, in Xi'an's demand prediction, UrbanMind surpassed UrbanGPT, further highlighting its ability to model complex spatial-temporal relationships and intercorrelations among dynamics. These results establish UrbanMind as a robust and effective model for urban traffic dynamics prediction.

Results of Question (3) (Ablation Study). We conducted an ablation study to evaluate the contribution of each component in UrbanMind, with results presented in Tab. 3. Removing Muffin-MAE leads to a significant drop in performance, with notably higher MAE and RMSE values across datasets, underscoring its importance in capturing intercorrelated spatial-temporal dependencies. Similarly, omitting LLM fine-tuning or test-time adaptation results in substantial performance degradation, highlighting their critical role in improving adaptability and precision, especially in zero-shot scenarios. Additionally, we examine the effects of different masking mechanisms in Muffin-MAE (*e.g.*, removing temporal, spatial, or global masking) and assess the impact of multifaceted and target embeddings in the final spatial-temporal tokens (*e.g.*, removing

Dataset	Metric	w/o Muffin-MAE	Muffin-MAE with different masks			Spatial-Temporal Token		w/o Fine-tuning	w/o Adaptation	UrbanMind
			w/o Temporal	w/o Spatial	w/o Global	w/o Target	w/o Multifaceted			
Speed	MAE	0.165	0.144	0.140	0.141	0.148	0.153	0.145	0.142	**0.137**
	RMSE	0.221	0.168	0.165	0.168	0.190	0.189	0.187	0.166	**0.162**
Inflow	MAE	0.250	0.162	0.175	0.180	0.189	0.220	0.143	0.124	**0.114**
	RMSE	0.343	0.201	0.205	0.224	0.247	0.293	0.209	0.218	**0.173**
Demand	MAE	0.235	0.191	0.219	0.220	0.202	0.245	0.214	0.179	**0.160**
	RMSE	0.389	0.300	0.327	0.328	0.380	0.378	0.345	0.283	**0.273**

Table 3: Ablation Study: Impacts of UrbanMind components (Muffin-MAE, LLM fine-tuning strategy, test time adaptation), masking mechanisms (temporal, spatial, global masking), and spatial-temporal token (including multifaceted and target embeddings) on zero-shot prediction accuracy. The analysis is conducted on Xi'an City.

Algorithm 1 Algorithm for UrbanMind

Input: Target urban dynamics $\mathcal{X}^k$, related multifaceted dynamics $\mathcal{X}$, initialized Muffin-MAE encoders E_{ϕ_1} and E_{ϕ_2}, decoders D_{ψ_1} and D_{ψ_2}, predictor P_θ, reconstructor G_τ, and pretrained LLM Q_ω.
Output: Well-trained E_{ϕ_1}, E_{ϕ_2}, D_{ψ_1}, D_{ψ_2}, Q_ω, P_θ and G_τ.
Step 1: Mask-Empowered Representation Learning:
for iteration $i = 1, 2, 3, \cdots$ **do**
  Sample a batch of $\mathcal{X}$ and $\mathcal{X}^k$.
  Train E_{ϕ_1}, E_{ϕ_2}, D_{ψ_1} and D_{ψ_2} with Eq. (1).
end for
Get the multifaceted embeddings $\mathcal{V}$ with E_{ϕ_1} and the target embeddings $\mathcal{V}^k$ with E_{ϕ_2}.
Get the final tokens by $\mathcal{U} = \text{concat}(\mathcal{V}, \mathcal{V}^k)$.
Step 2: Semantic-Aware Prompting and Fine-Tuning:
for iteration $= 1, 2, 3, \cdots$ **do**
  Combine $\mathcal{U}$ with text descriptions as prompts.
  Fine-tune Q_ω and update P_θ with prompts using Eq.(2).
end for
Step 3: Test Time Adaptation and Final Prediction:
for iteration $i = 1, 2, 3, \cdots$ **do**
  Sample a testing sequence $\mathcal{X}^{\text{ts}}$, and produce the a latent embedding sequence $\mathcal{E}$ using Q_ω.
  Get $\mathcal{E}^{\text{masked}}$ with Eq.(3) as the input of G_τ.
  Update G_τ with Eq.(4).
end for
Produce final prediction with $P_\theta(\mathcal{X}^{\text{ts}})$.

either target embeddings or multifaceted embeddings) on prediction accuracy. In both cases, performance degradation is evident in increased MAE and RMSE, further validating the effectiveness of these components in UrbanMind.

Results of Question (4) (Hyperparameters Evaluation). To analyze the impact of different hyperparameters on model performance, we conducted experiments on traffic speed prediction in Xi'an, focusing on four key hyperparameters: the temporal masking ratio p_t, spatial masking ratio p_s, number of trainable layers during LLM fine-tuning, and number of multifaceted dynamics. The results are presented in Fig. 5. For p_t, as shown in Fig. 5(a), increasing the ratio to 0.33 achieves the lowest RMSE, while both lower and higher masking ratios degrade performance. Similarly, Fig. 5(b) demonstrates that the spatial masking ratio p_s achieves optimal performance at $p_s = 0.25$, with performance gradually worsening as the ratio increases, suggesting that excessive spatial masking hinders learning. The effect of trainable layers during LLM fine-tuning, shown in Fig. 5(c), reveals that increasing the number of fine-tuned layers generally reduces RMSE, with a slight fluctuation at 4 layers, while overall, more layers help the model capture complex spatial-temporal relationships more effectively. Finally, Fig. 5(d) illustrates that incorporating more multifaceted dynamics consistently improves RMSE, highlighting the importance of including multifaceted embeddings in Muffin-MAE in capturing intercorrelated spatial-temporal relationships.

5 Related Work

Spatial-temporal urban prediction is essential for effective city management. Recent advancements have leveraged deep learning techniques to enhance predictive accuracy in diverse urban domains. Models such as graph neural networks [11, 17, 44, 45, 58], Transformers [12, 13, 16, 22, 36], and generative adversarial networks [51, 53, 55], have been introduced to capture spatial-temporal patterns. Additionally, cutting-edge techniques like contrastive learning [15, 18, 29], and diffusion models [27, 34, 41, 47] have further enhanced the prediction performance. Despite these innovations, most existing approaches are constrained by the need to train separate models for each specific dataset, limiting their adaptability and scalability. While some studies have explored transfer learning and meta-learning across regions [14, 48, 49, 54, 57], these methods still rely on data from the target region. In contrast, our proposed model addresses these limitations by integrating spatial-temporal data with Large Language Models, offering a generalized solution capable of adapting across diverse urban scenarios.

Large Language Models (LLMs) have revolutionized artificial intelligence, driving breakthroughs in text comprehension, reasoning, and generalization across diverse domains. Notable models such as ChatGPT [28], Claude [30], LLaMA [40], Vicuna [5], and ChatGLM [7] have gained widespread attention due to their scalability and adaptability, driving research into their diverse applications. These models have demonstrated exceptional capabilities in tasks such as graph-based reasoning [31] and multimodal learning [1], showcasing their potential to extend beyond traditional text-centric applications. Furthermore, their integration with advanced strategies like few-shot learning [2] has opened new avenues for addressing challenges in data-scarce domains. Despite these advancements, the application of LLMs to zero-shot and few-shot learning in urban spatial-temporal prediction remains nascent, constrained by challenges such as adapting domain-specific data and addressing interdependencies across spatial-temporal scales.

Spatial-Temporal Pretraining Models have emerged as a promising approach to enhancing predictive performance in urban scenarios. Models such as UrbanGPT [19] and UniST [46] integrate spatial-temporal data with LLMs using prompt tuning. Other works, including GPT-ST [21], ST-LLM [23], TPLLM [32], GATGPT [3], and STG-LLM [25], primarily focus on graphs or grid-based worlds, adapting single spatial-temporal datasets for use with LLMs. However, these models are either ill-equipped to handle the inter-correlated multi-faceted dynamics inherent in urban environments or rely solely on the inherent generalization ability of LLMs to adapt across different scenarios, overlooking the critical issue of distributional shifts between training and testing data.

6 Conclusion

In this paper, we present UrbanMind, a novel spatial-temporal LLM designed to effectively process and understand the intricate relationships and patterns within spatial-temporal data for urban dynamics prediction. UrbanMind demonstrates high prediction accuracy and robust generalization, including in zero-shot scenarios where no prior data is available. Its key innovations include a masked-empowered representation learning framework implemented through the novel Muffin-MAE, which employs advanced masking strategies to capture complex spatial-temporal dependencies and inter-correlations across multi-faceted urban dynamics. Additionally, UrbanMind incorporates a semantic-rich prompt design and fine-tuning strategy tailored for spatial-temporal data, as well as a testing-time adaptation mechanism that mitigates distributional shifts through a reconstruction module. Extensive experiments on three urban dynamics—traffic speed, inflow, and travel demand—across three cities validate UrbanMind's effectiveness. The results highlight its superior generalization and predictive accuracy across diverse spatial-temporal scenarios, consistently outperforming state-of-the-art baselines, even in unseen regions or data-scarce settings.

7 Acknowledgments

Yanhua Li was supported in part by NSF grants IIS-1942680 (CAREER), CNS-1952085 and DGE-2021871. Jun Luo is supported by The Innovation and Technology Fund (Ref. ITP/012/25LP).

Disclaimer: Any opinions, findings, conclusions or recommendations expressed in this material/event (or by members of the project team) do not reflect the views of the Government of the Hong Kong Special Administrative Region, the Innovation and Technology Commission or the Innovation and Technology Fund Research Projects Assessment Panel.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a Visual Language Model for Few-Shot Learning. *Advances in neural information processing systems* 35 (2022), 23716–23736.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems* 33 (2020), 1877–1901.

[3] Yakun Chen, Xianzhi Wang, and Guandong Xu. 2023. GATGPT: A Pre-trained Large Language Model with Graph Attention Network for Spatiotemporal Imputation. arXiv:2311.14332 [cs.LG] https://arxiv.org/abs/2311.14332

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

[5] Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)* 2, 3 (2023), 6.

[6] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. 2019. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems* 21, 11 (2019), 4883–4894.

[7] Zhengxiao Du, Yujie Qian, Xiao Liu, et al. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. *arXiv preprint arXiv:2203.02155* (2022).

[8] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[9] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. 2022. Masked Autoencoders As Spatiotemporal Learners. arXiv:2205.09113 [cs.CV] https://arxiv.org/abs/2205.09113

[10] Junfeng Hu, Xu Liu, Zhencheng Fan, Yuxuan Liang, and Roger Zimmermann. 2024. Towards unifying diffusion models for probabilistic spatio-temporal graph learning. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*. 135–146.

[11] Rongzhou Huang, Chuyin Huang, Yubao Liu, Genan Dai, and Weiyang Kong. 2020. LSGCN: Long Short-Term Traffic Prediction with Graph Convolutional Networks. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 2355–2361. https://doi.org/10.24963/ijcai.2020/326 Main track.

[12] Songtao Huang, Hongjin Song, Tianqi Jiang, Akbar Telikani, Jun Shen, Qingguo Zhou, Binbin Yong, and Qiang Wu. 2024. DST-GTN: Dynamic Spatio-Temporal Graph Transformer Network for Traffic Forecasting. *arXiv preprint arXiv:2404.11996* (2024).

[13] Guanyao Li, Shuhan Zhong, S.-H. Gary Chan, Ruiyuan Li, Chih-Chieh Hung, and Wen-Chih Peng. 2022. A Lightweight and Accurate Spatial-Temporal Transformer for Traffic Forecasting. *arXiv preprint arXiv:2201.00008* (2022).

[14] Ke Li, Yanjie Zhu, Chao Zhang, and Zhenhui Li. 2022. Selective Cross-City Transfer Learning for Traffic Prediction via Source Domain Mixing. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. 2014–2024.

[15] Lincan Li, Kaixiang Yang, Fengji Luo, and Jichao Bi. 2023. STS-CCL: Spatial-Temporal Synchronous Contextual Contrastive Learning for Urban Traffic Forecasting. *arXiv preprint arXiv:2307.02507* (2023).

[16] Y. Li et al. 2024. TST-Trans: A Transformer Network for Urban Traffic Flow Prediction. *IEEE Transactions on Intelligent Transportation Systems* (2024).

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

[18] Zechen Li, Weiming Huang, Kai Zhao, Min Yang, Yongshun Gong, and Meng Chen. 2023. Urban Region Embedding via Multi-View Contrastive Prediction. *arXiv preprint arXiv:2312.09681* (2023).

[19] 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* (Barcelona, Spain) *(KDD '24)*. 5351–5362. https://doi.org/10.1145/3637528.3671578

[20] 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*. 5351–5362.

[21] Zhonghang Li, Lianghao Xia, Yong Xu, and Chao Huang. 2023. GPT-ST: Generative Pre-Training of Spatio-Temporal Graph Neural Networks. *arXiv preprint arXiv:2311.04245* (2023).

[22] Jiaqi Lin and Qianqian Ren. 2024. Rethinking Spatio-Temporal Transformer for Traffic Prediction: Multi-level Multi-view Augmented Learning Framework.

arXiv preprint arXiv:2406.11921 (2024).
[23] Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao. 2024. Spatial-Temporal Large Language Model for Traffic Prediction. arXiv:2401.10134 [cs.LG] https://arxiv.org/abs/2401.10134
[24] Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao. 2024. Spatial-temporal large language model for traffic prediction. *arXiv preprint arXiv:2401.10134* (2024).
[25] Lei Liu, Shuo Yu, Runze Wang, Zhenxun Ma, and Yanming Shen. 2024. How Can Large Language Models Understand Spatial-Temporal Data? arXiv:2401.14192 [cs.LG] https://arxiv.org/abs/2401.14192
[26] Lei Liu, Shuo Yu, Runze Wang, Zhenxun Ma, and Yanming Shen. 2024. How can large language models understand spatial-temporal data? *arXiv preprint arXiv:2401.14192* (2024).
[27] Yuhang Liu, Yingxue Zhang, Xin Zhang, Yu Yang, Yiqun Xie, Sahar Ghanipoor Machiani, Yanhua Li, and Jun Luo. 2024. Align Along Time and Space: A Graph Latent Diffusion Model for Traffic Dynamics Prediction. In *2024 IEEE International Conference on Data Mining (ICDM)*. IEEE, 271–280.
[28] OpenAI. 2023. ChatGPT: Optimizing Language Models for Dialogue. *OpenAI White Paper* (2023). https://openai.com/chatgpt
[29] Lin Pan, Qianqian Ren, Zilong Li, and Xingfeng Lv. 2025. Rethinking Spatial-Temporal Contrastive Learning for Urban Traffic Flow Forecasting: Multi-Level Augmentation Framework. *Complex & Intelligent Systems* 11, 1 (2025), 96.
[30] Aman Priyanshu, Yash Maurya, and Zuofei Hong. 2024. AI Governance and Accountability: An Analysis of Anthropic's Claude. *arXiv preprint arXiv:2407.01557* (2024).
[31] Guanchu Qian, Pan Zhang, and Jundong Li. 2023. Large Language Models for Graph-Based Reasoning. *arXiv preprint arXiv:2305.02340* (2023).
[32] Yilong Ren, Yue Chen, Shuai Liu, Boyue Wang, Haiyang Yu, and Zhiyong Cui. 2024. TPLLM: A Traffic Prediction Framework Based on Pretrained Large Language Models. arXiv:2403.02221 [cs.LG] https://arxiv.org/abs/2403.02221
[33] Yilong Ren, Yue Chen, Shuai Liu, Boyue Wang, Haiyang Yu, and Zhiyong Cui. 2024. TPLLM: A traffic prediction framework based on pretrained large language models. *arXiv preprint arXiv:2403.02221* (2024).
[34] Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. 2023. DYffusion: A Dynamics-informed Diffusion Model for Spatiotemporal Forecasting. *Advances in neural information processing systems* 36 (2023), 45259–45287.
[35] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part I 25*. Springer, 362–373.
[36] Zhiqi Shao, Michael G. H. Bell, Ze Wang, D. Glenn Geers, Xusheng Yao, and Junbin Gao. 2024. CCDSReFormer: Traffic Flow Prediction with a Criss-Crossed Dual-Stream Enhanced Rectified Transformer Model. *arXiv preprint arXiv:2403.17753* (2024).
[37] Yu Sun, Xinhao Li, Karan Dalal, Chloe Hsu, Sanmi Koyejo, Carlos Guestrin, Xiaolong Wang, Tatsunori Hashimoto, and Xinlei Chen. 2023. Learning to (learn at test time). *arXiv preprint arXiv:2310.13807* (2023).
[38] Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. 2024. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620* (2024).
[39] Mingtian Tan, Mike Merrill, Vinayak Gupta, Tim Althoff, and Tom Hartvigsen. 2024. Are language models actually useful for time series forecasting? *Advances in Neural Information Processing Systems* 37 (2024), 60162–60191.
[40] Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971* (2023).
[41] Zeyu Wang, Zecheng Hao, Jingyu Lin, Yuchao Feng, and Yufei Guo. 2024. UP-Diff: Latent Diffusion Model for Remote Sensing Urban Prediction. *arXiv preprint arXiv:2407.11578* (2024).
[42] Mengran Xia, Dawei Jin, and Jingyu Chen. 2022. Short-term traffic flow prediction based on graph convolutional networks and federated learning. *IEEE Transactions on Intelligent Transportation Systems* 24, 1 (2022), 1191–1203.
[43] Hanyi Yang, Lili Du, Guohui Zhang, and Tianwei Ma. 2023. A Traffic Flow Dependency and Dynamics based Deep Learning Aided Approach for Network-Wide Traffic Speed Propagation Prediction. *Transportation Research Part B: Methodological* 167 (2023), 99–117. https://doi.org/10.1016/j.trb.2022.11.009
[44] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875* (2017).
[45] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-2018)*. International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2018/505
[46] Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. 2024. UniST: A Prompt-Empowered Universal Model for Urban Spatio-Temporal Prediction. *arXiv preprint arXiv:2402.11838* (2024).
[47] Chengyang Zhang, Yong Zhang, Qitan Shao, Bo Li, Yisheng Lv, Xinglin Piao, and Baocai Yin. 2023. ChatTraffic: Text-to-Traffic Generation via Diffusion Model. *arXiv preprint arXiv:2311.16203* (2023).
[48] Junbo Zhang, Yu Zheng, and Dong Qi. 2019. Cross-City Transfer Learning for Deep Spatio-Temporal Prediction. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*. 2827–2833.
[49] Junbo Zhang, Yu Zheng, and Dong Qi. 2022. When Transfer Learning Meets Cross-City Urban Flow Prediction: A Deep Spatio-Temporal Network with Attention Mechanism. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*. 2030–2036.
[50] Xin Zhang, Yanhua Li, Xun Zhou, Oren Mangoubi, Ziming Zhang, Vincent Filardi, and Jun Luo. 2021. Dac-ml: domain adaptable continuous meta-learning for urban dynamics prediction. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 906–915.
[51] Yingxue Zhang, Yanhua Li, Xun Zhou, Xiangnan Kong, and Jun Luo. 2019. TrafficGAN: Off-Deployment Traffic Estimation with Traffic Generative Adversarial Networks. In *ICDM*.
[52] Yingxue Zhang, Yanhua Li, Xun Zhou, Xiangnan Kong, and Jun Luo. 2019. TrafficGAN: Off-Deployment Traffic Estimation with Traffic Generative Adversarial Networks. In *2019 IEEE International Conference on Data Mining (ICDM)*. 1474–1479. https://doi.org/10.1109/ICDM.2019.00193
[53] Yingxue Zhang, Yanhua Li, Xun Zhou, Xiangnan Kong, and Jun Luo. 2020. CurbGAN: Conditional Urban Traffic Estimation through Spatio-Temporal Generative Adversarial Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 842–852. https://doi.org/10.1145/3394486.3403127
[54] Yingxue Zhang, Yanhua Li, Xun Zhou, Xiangnan Kong, and Jun Luo. 2022. STransGAN: Spatially-Transferable Generative Adversarial Networks for Urban Traffic Estimation. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 743–752.
[55] Yingxue Zhang, Yanhua Li, Xun Zhou, Zhenming Liu, and Jun Luo. 2021. C3-GAN: Complex-Condition-Controlled Urban Traffic Estimation through Generative Adversarial Networks. In *2021 IEEE International Co10.1016/j.knosys.2023.110591nference on Data Mining (ICDM)*. 1505–1510. https://doi.org/10.1109/ICDM51629.2021.00196
[56] Yingxue Zhang, Yanhua Li, Xun Zhou, and Jun Luo. 2020. cST-ML: Continuous Spatial-Temporal Meta-Learning for Traffic Dynamics Prediction. In *International Conference on Data Mining*.
[57] Yingxue Zhang, Yanhua Li, Xun Zhou, and Jun Luo. 2022. Mest-GAN: Cross-City Urban Traffic Estimation with Me ta S patial-T emporal G enerative A dversarial N etworks. In *2022 IEEE International Conference on Data Mining (ICDM)*. 733–742. https://doi.org/10.1109/ICDM54844.2022.00084
[58] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. GMAN: A Graph Multi-Attention Network for Traffic Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 01 (Apr. 2020), 1234–1241. https://doi.org/10.1609/aaai.v34i01.5477
[59] Xiaobei Zou, Luolin Xiong, Yang Tang, and Jürgen Kurths. 2024. SAMSGL: Series-aligned multi-scale graph learning for spatiotemporal forecasting. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 34, 6 (2024).

A APPENDIX FOR REPRODUCIBILITY

To support the reproducibility of the results in this paper, we publish our code and data [2]. Here, we describe the dataset and baseline settings in detail.

A.1 Detailed Description of the dataset

As previously mentioned, the dataset consists of three types of data: (1) traffic speed, (2) taxi inflow, and (3) travel demand. These data are derived from urban records collected in Shenzhen, Xi'an, and Chengdu, China. The details of the original dataset are presented in Tab. 4.

Data Preprocessing for Shenzhen: To expand our datasets, we applied a basic data augmentation technique by segmenting large city maps into smaller regions. Using Shenzhen as an example, the city is mapped onto a 40×50 grid, which is then subdivided into smaller regions R_{ij}, each consisting of 10×10 grid cells. Starting with region R_{11}, we extract the initial 10×10 section. We then shift

[2] UrbanMind code: https://doi.org/10.5281/zenodo.15484938

the window 5 grid cells to the right (from s_{11} to s_{16}) to obtain the next region, R_{12}. By continuously sliding the window in this manner, we generate multiple overlapping regions R_{ij}, all maintaining the same 10×10 size. The final dataset for Shenzhen has dimensions of (162, 12, 63, 10, 10), corresponding to the number of days, time slots per day, number of regions, and the width and height of each region.

Data Preprocessing for Xi'an and Chengdu: To process the datasets for Xi'an and Chengdu, the urban areas are divided into 20×20 grid cells, which are further partitioned into four regions R_{ij} without augmentation, each consisting of 10×10 grid cells. Specifically, the grid is evenly split along both horizontal and vertical axes, resulting in regions R_{11}, R_{12}, R_{21}, and R_{22}. The final dataset for both Xi'an and Chengdu has a shape of (30, 12, 4, 10, 10), representing (number of days, time slots per day, number of regions, region width, region height). We train on 3 regions and select the remaining region for testing.

Normalization: To address outliers in the datasets from all three cities, we implemented tailored strategies for each data type. For the traffic speed data, we imposed an upper limit of 140, with any values above this threshold clipped to 140. In the case of taxi inflow and travel demand, we defined the upper bounds based on the 90th percentile of their respective data distributions to effectively manage extreme values. Following the outlier treatment, all datasets underwent min-max normalization, scaling the data to a range between -1 and 1.

Table 4: Dataset descriptions.

City	City size	Data	Timespan
Shenzhen	40 × 50	Speed	07/01/16-12/31/16
		Inflow	07/01/16-12/31/16
		Demand	07/01/16-12/31/16
Chengdu	20 × 20	Speed	10/01/16-10/31/16
		Inflow	10/01/16-10/31/16
		Demand	10/01/16-10/31/16
Xi'an	20 × 20	Speed	10/01/16-10/31/16
		Inflow	10/01/16-10/31/16
		Demand	10/01/16-10/31/16

Data Description: We evaluate our model using nine real-world urban dynamics datasets, covering three data types: (1) traffic speed, (2) taxi inflow, and (3) travel demand. Three datasets were collected in Shenzhen, China, from July 1 to December 31, 2016. The city is divided into a 40 × 50 grid, which is further aggregated into 63 regions, each consisting of 10 × 10 grid cells, covering the entire city. The other six datasets come from Xi'an and Chengdu, China, spanning October 1 to October 31, 2016. Both cities are partitioned into 20×20 grids and divided into 4 regions of 10×10 grid cells each, where traffic speed, taxi inflow, and travel demand are measured.

Here are the details of the urban dynamics datasets:

- **Traffic Speed.** In Shenzhen, the dataset includes traffic speed data for 63 regions over 4416 one-hour time slots spanning 6 months. For Xi'an and Chengdu, it covers 4 regions with 360 one-hour time slots over 1 month. Traffic speed is calculated as the ratio of travel distance to travel time for each grid cell per time slot.
- **Taxi Inflow.** The Shenzhen dataset records taxi inflow for 63 regions across 4416 one-hour time slots, while the Xi'an and Chengdu datasets cover 4 regions with 360 time slots each. Taxi inflow represents the number of taxi arrivals at each grid cell within a specific hour.
- **Travel Demand.** This dataset tracks travel demand in 63 regions of Shenzhen (4416 time slots) and 4 regions of Xi'an and Chengdu (360 time slots each). Travel demand is measured by the number of taxi pickups and drop-offs within each grid cell per hour, serving as a proxy for overall demand based on taxi data, as validated in prior studies [52–55].

In short, the three datasets from Shenzhen have dimensions of $(162 \times 63, 12, 10, 10)$, where 162 represents the number of valid days, 63 is the number of regions, 12 indicates one-hour time slots per day, and 10×10 is the region size. Similarly, the six datasets from Xi'an and Chengdu are sized $(30 \times 4, 12, 10, 10)$, with 30 valid days, 4 regions, 12 one-hour time slots per day, and 10×10 grid cells per region.

A.2 Baselines Settings

We used the Adam optimizer in both the Mask-Empowered Representation Learning and Semantic-Aware Prompting stages, with a learning rate of 0.00001 for Shenzhen and 0.0001 for Xi'an and Chengdu. The same optimizer was applied during Test Time Adaptation, using learning rates of 0.00005 for Shenzhen and 0.0005 for the other two cities. UrbanMind is built on the LLaMA3 backbone. Detailed architectures and settings of the baselines are provided below.

- **GATGPT[4]**: GATGPT combines a pre-trained GPT-2 model with a graph attention layer to model spatial-temporal dependencies in traffic prediction. The graph attention layer captures spatial relationships using adaptive weights from region adjacency, and its output is fed into GPT-2 to learn temporal patterns. A final linear layer maps the outputs to the target spatial grid.
- **GCNGPT[4, 24]**: GCNGPT integrates graph convolutional networks (GCNs) with a pre-trained GPT-2 model to effectively capture spatial-temporal dependencies in traffic data. The model first applies a GCN to extract spatial features from region-wise adjacency matrices, transforming the input into a representation that encodes spatial relationships. This output is then reshaped and fed into the GPT-2 model to model complex temporal dynamics. A linear projection layer maps the GPT-2 outputs to the target grid for final predictions.
- **STLLM[24]**: ST-LLM uses large language models for traffic prediction by combining spatial-temporal embeddings with a partially frozen attention strategy to capture global dependencies. The baseline integrates spatial and temporal features to enhance traffic dynamics representation, leveraging LLMs to model complex patterns. Key attention layers are partially frozen to ensure stability while allowing selective fine-tuning for traffic-specific tasks. The model processes inputs with embedded time features, and predictions are made through a transformer-based architecture optimized with mean squared error loss.
- **TPLLM[33]**: TP-LLM integrates temporal and spatial information for traffic prediction using a hybrid approach that combines sequence embeddings, graph embeddings, and a

GPT-2 backbone. The model starts by transforming the input data through a sequence embedding layer using 1D convolutions and a graph embedding layer via linear transformations to capture temporal patterns and spatial relationships, respectively. These embeddings are then fused and passed through a GPT-2 model configured with six layers and eight attention heads, enabling the extraction of complex temporal dependencies. The output from GPT-2 is processed through a linear layer to map it to the desired prediction dimensions.

- **UrbanGPT[20]**: UrbanGPT integrates a spatio-temporal dependency encoder with instruction-tuning to enable large language models (LLMs) to handle complex time-space interdependencies and generalize across diverse urban tasks. The baseline model employs the spatial-temporal encoder (ST_Enc) in conjunction with LLaMA2[40] to capture dynamic patterns in urban data, effectively modeling both spatial relationships and temporal trends. The ST_Enc module incorporates dilated inception convolution layers to extract multi-scale spatiotemporal features, which are then processed through LLaMA2 to capture global dependencies. Instruction-tuning enhances the model's adaptability.
- **STG-LLM[26]**: STG-LLM integrates a spatial-temporal graph tokenizer with a lightweight adapter to process complex spatial-temporal data effectively. The model begins with a linear encoder that transforms raw spatial-temporal inputs into a latent representation. Temporal and weekly patterns are captured through time-of-day and day-of-week embedding layers, which provide temporal positional context. These encoded features are then combined with positional embeddings and passed through a Transformer encoder, simulating an GPT-2 to model long-range dependencies across time and space. Finally, a linear decoder refines the output.
- **LLaMA3[8]**: LLaMA3 is a foundation model designed for multilingual understanding and reasoning. Its advanced capabilities make it ideal for traffic dynamics prediction. The baseline model leverages LLaMA3 to capture complex temporal dependencies in spatial-temporal data. The model architecture integrates LLaMA3 as a feature extractor, with its parameters frozen except for select layers to maintain efficiency. The extracted features are processed through a linear layer to predict future traffic conditions, reshaping the output to match the spatial grid structure. Specifically, we utilized LLaMA 3.2, a foundational large language model developed by Meta, as the baseline model for UrbanMind.
- **DYffusion[34]** DYffusion is a diffusion model designed for probabilistic spatial-temporal forecasting, aiming to generate stable and accurate rollout predictions. The baseline integrates a Unet architecture with a Gaussian Diffusion process to improve performance. The Unet processes single-channel inputs through an initial convolutional layer and uses multiple convolutional layers with scaled dimensions. The Gaussian Diffusion module refines predictions over time through iterative denoising steps.
- **TGC-LSTM[6]**: TGC-LSTM addresses spatial-temporal forecasting in traffic networks by modeling time-varying patterns and complex spatial dependencies. It combines traffic graph convolution and spectral graph convolution within an LSTM framework to capture both spatial and temporal dynamics. The LSTM models temporal dependencies, while convolutional layers extract spatial features from the traffic network. The model flattens spatial dimensions for LSTM processing and reconstructs outputs to retain spatial structure. It is trained using mean squared error loss.
- **GCRN[35]**: GCRN extends classical RNNs to handle arbitrary graph-structured data, enabling effective modeling of spatial and temporal dependencies. The baseline model integrates a graph convolutional layer with a gated recurrent unit (GRU) to capture both spatial relationships among nodes and temporal dynamics over time. The graph convolutional layer processes input features to learn spatial dependencies based on the adjacency matrix, while the GRU captures temporal patterns from sequential data. The final output is generated through a fully connected layer that predicts future traffic conditions.
- **GAGCN[42]**: GAGCN uses graph attention networks to extract and dynamically adjust spatial associations among nodes over time. The baseline builds on this by integrating graph attention into a convolutional neural network to capture spatial-temporal dependencies. Input traffic data is processed through stacked convolutional layers to learn local spatial features, followed by graph attention layers that refine node relationships based on temporal patterns. Final predictions are made through fully connected layers, and the model is optimized with mean squared error loss.

B Appendix B: Additional Experiments

B.1 Computational Efficiency Details

We report the runtime cost of our LLaMA3-based experiments on the Shenzhen dataset. UrbanMind takes 70.9 seconds per epoch for training and 16.5 seconds per epoch for test-time adaptation. In comparison, the baseline UrbanGPT (also based on LLaMA) requires 80.1 seconds per epoch for training. Although UrbanMind introduces moderate additional cost from Muffin-MAE and the adaptation module, it remains practical and suitable for real-time inference scenarios, especially considering its improved generalization capability.

B.2 Cross-City Generalization

To evaluate cross-city generalization, we conduct a zero-shot transfer experiment where models are trained on traffic speed data from Shenzhen and directly evaluated in Xi'an without fine-tuning. We specifically compare UrbanMind against UrbanGPT. UrbanMind achieves 8.5% lower MAE (0.194 vs. 0.212) and 9.9% lower RMSE (0.236 vs. 0.262), demonstrating its superior adaptability to unseen urban environments.

B.3 Additional Baseline

We include STGAIL [27] as a new baseline on traffic speed prediction in Xi'an. STGAIL achieves an MAE of 0.227 and RMSE of 0.266, underperforming compared to UrbanMind (0.194 / 0.236). This suggests its limited capacity for handling multimodal spatial-temporal signals in urban forecasting.