OpenCity: Open Spatio-Temporal Foundation Models for Traffic Prediction

Zhonghang Li[1,2], Long Xia[3], Lei Shi[3], Yong Xu[2], Dawei Yin[3] and Chao Huang[1*]
[1]The University of Hong Kong, [2]South China University of Technology, [3]Baidu Inc.

ABSTRACT

Accurate traffic forecasting is crucial for effective urban planning and transportation management, enabling efficient resource allocation and enhanced travel experiences. However, existing models often face limitations in generalization, struggling with zero-shot prediction on unseen regions and cities, as well as diminished long-term accuracy. This is primarily due to the inherent challenges in handling the spatial and temporal heterogeneity of traffic data, coupled with the significant distribution shift across time and space. In this work, we aim to unlock new possibilities for building versatile, resilient and adaptive spatio-temporal foundation models for traffic prediction. To achieve this goal, we introduce a novel foundation model, named OpenCity, that can effectively capture and normalize the underlying spatio-temporal patterns from diverse data characteristics, facilitating zero-shot generalization across diverse urban environments. OpenCity integrates the Transformer architecture with graph neural networks to model the complex spatio-temporal dependencies in traffic data. By pre-training OpenCity on large-scale, heterogeneous traffic datasets, we enable the model to learn rich, generalizable representations that can be seamlessly applied to a wide range of traffic forecasting scenarios. Experimental results demonstrate that OpenCity exhibits exceptional zero-shot predictive performance. Moreover, OpenCity showcases promising scaling laws, suggesting the potential for developing a truly one-for-all traffic prediction solution that can adapt to new urban contexts with minimal overhead. We made our proposed OpenCity model open-source and it is available at the following link: https://github.com/HKUDS/OpenCity.

ACM Reference Format:
Zhonghang Li[1,2], Long Xia[3], Lei Shi[3], Yong Xu[2], Dawei Yin[3] and Chao Huang[1*]. 2024. OpenCity: Open Spatio-Temporal Foundation Models for Traffic Prediction. In *Proceedings of ACM (Conference).* ACM , 12 pages. https://doi.org/10.1145/nnnnnnn.

1 INTRODUCTION

Transportation is an essential component of urban activities, serving as the fundamental infrastructure that supports the efficient movement of people and goods within cities [33, 41]. Precise traffic forecasting allows for proactive traffic management, enabling transportation planners to anticipate and mitigate potential congestion, delays, and disruptions [13, 17]. This, in turn, enhances the overall efficiency and sustainability of urban transportation networks, fostering the real-world intelligent transportation systems.

The success of deep learning has empowered spatio-temporal models for traffic forecasting. These models leverage deep neural networks to learn effective representations that can better capture the spatial and temporal dependencies inherent in urban traffic data. However, despite these advancements, current traffic prediction models often face significant limitations in terms of generalization.

Cross-Regional Model Spatial Generalization. Firstly, A key limitation of current traffic prediction models is their struggle with spatial generalization. These models often fail to perform well when applied to unseen regions or cities, as traffic patterns and dynamics can vary considerably across geographical locations due to factors like infrastructure, demographics, and urban planning [1]. Existing models typically learn from data limited to specific regions, making them unable to effectively generalize their knowledge to capture the unique characteristics of unfamiliar traffic environments.

This issue is critical, as it is impractical to deploy comprehensive sensor networks across entire urban areas to collect traffic data [15, 26]. A more viable approach is to build models that can generalize well to unseen regions using only partial data. Additionally, developing spatio-temporal models that are applicable across different cities would significantly reduce deployment and maintenance costs [15, 40]. However, when these models are applied in new locations, they often experience a significant drop in predictive performance, hindering their wider applicability. This spatial generalization challenge is crucial to address in order to create traffic forecasting solutions that can be seamlessly deployed across diverse urban settings without extensive retraining or fine-tuning.

Temporal Generalization for Long-term Forecasting. Current traffic prediction models excel at short-term forecasting, such as anticipating conditions for the next hour [3, 12]. However, their ability to generalize to longer time frames, like days or weeks ahead, is notably limited. This limitation is largely due to the models' poor generalization ability in effectively handling the evolving temporal distribution shifts that occur over longer time horizons in practical urban scenarios. As the forecasting timeframe increases, these models struggle to capture and account for the dynamic changes in traffic patterns that influence long-term traffic conditions.

This limitation presents a considerable obstacle for city planners and transportation agencies striving to devise effective long-term strategies. Precise long-term traffic forecasts are crucial for a range of applications, including proactive infrastructure planning, public transit scheduling, and event logistics coordination. Despite their importance, existing models often fall short when it comes to providing reliable predictions beyond the immediate future.

In recent years, the remarkable capabilities of large foundation models have captured significant attention across various domains. These models, trained on vast and diverse datasets, have demonstrated exceptional performance in natural language processing (NLP) [2, 28] and computer vision (CV) [10, 16] tasks, showcasing their remarkable abilities to comprehend and generalize from complex data. While the success of foundation models has been

[*] Chao Huang is the Corresponding Author.

Conference, 2024,
2024. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.

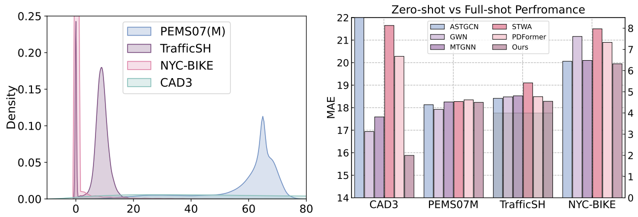

Figure 1: The left figure showcases varying data distributions across traffic datasets, highlighting the need for a model that can handle such disparities. The right figure compares the zero-shot OpenCity performance to full-shot baselines, showing OpenCity performs comparably despite the distribution shift challenges from spatio-temporal heterogeneity.

well-documented in the NLP and CV realms, the development of a foundation model tailored specifically for urban traffic data remains largely unexplored. In this study, we aim to bridge this gap by creating a novel foundation model that is specifically designed to capture the complexities of urban traffic. Our primary focus is on enhancing the model's ability to generalize across both spatial and temporal dimensions, enabling it to perform robust cross-regional traffic forecasting and long-term temporal predictions.

Contribution. In this work, we introduce the One-for-All Spatio-Temporal Model-OpenCity, a novel and versatile framework for developing foundational urban traffic models with powerful zero-shot learning capabilities. The primary objective of OpenCity is to address the limitations of existing traffic forecasting approaches, which often struggle to generalize beyond their initial training domains. By providing a unified modeling architecture, OpenCity aims to enable seamless adaptation across a wide range of traffic conditions and contexts. Achieving this goal of versatility and adaptability, however, presents several critical challenges:

- **Learning Effective Universal Representations**. The ability to learn powerful, generalizable representations of traffic dynamics is crucial for developing a versatile traffic model. This challenge entails capturing the essential patterns and relationships underlying urban mobility, while minimizing the reliance on context-specific features that may impede the model's adaptability to new environments. By learning universal representations, the model can acquire a robust and transferable understanding of traffic patterns, enabling it to be effectively applied to a diverse range of scenarios, even without domain-specific training data.
- **Confronting Spatio-Temporal Heterogeneity**. Urban traffic patterns are characterized by their inherent diversity, showing notable variations in distribution across various spatial areas and timeframes. Addressing this heterogeneity is essential for ensuring that a unified spatio-temporal traffic model remains versatile and adaptable. For a model to seamlessly adjust to the multitude of traffic scenarios it may encounter, it must be crafted to adeptly manage these shifts in distribution.

To address the key challenges, the proposed model, OpenCity, integrates the Transformer architecture and graph neural networks. This hybrid approach enables the model to effectively capture the complex spatio-temporal dependencies inherent in traffic data. At the core of OpenCity is the seamless cooperation between three key components: a zero-shot spatio-temporal embedding layer, a spatio-temporal context encoder, and spatio-temporal dependency modeling components. This architecture allows the model to learn universal representations that can generalize across diverse spatial and temporal contexts. Furthermore, the model is strengthened through pre-training on large-scale, heterogeneous traffic datasets, equipping it with rich and transferable representations. In summary, this work makes the following major contributions:

- **Versatile Spatio-Temporal Modeling**. OpenCity is designed to effectively handle the inherent diversity and variations in urban traffic patterns across different spatial areas and timeframes. By confronting this spatio-temporal heterogeneity, OpenCity overcomes the reliance on context-specific features that often hinders the generalization capabilities of current spatio-temporal models.
- **Impressive Zero-Shot Forecasting Capabilities**. The OpenCity demonstrates superior performance compared to **full-shot** models trained exclusively on target domains. This remarkable **zero-shot** capability highlights the model's ability to learn generalizable representations, enabling seamless application to new traffic environments without extensive retraining or fine-tuning.
- **Fast Context-Adaptive Ability**. The remarkable adaptability of our OpenCity suggests its broad applicability across a diverse range of spatio-temporal prediction tasks. By requiring only efficient and expedited fine-tuning for context adaptation, the model can be seamlessly deployed in a wide variety of scenarios.
- **Scaling Laws Investigation**. OpenCity showcases promising scaling laws, suggesting the potential for effectively scaling and adapting to new, previously unseen scenarios with minimal additional training or fine-tuning required across diverse contexts.

2 PRELIMINARIES

Spatio and Temporal Unit Generation. Traffic data is inherently spatio-temporal in nature, reflecting the dynamic patterns and spatial distribution of transportation networks. It is typically represented as a two-dimensional matrix $\mathbf{X} \in \mathbb{R}^{R \times T}$, where each element corresponds to a specific traffic metric (*e.g.*, flow, speed, or demand), for a given region r and time interval t. There are two prevalent forms of spatial region unit used to model traffic data:

- i) **Sensor-based Traffic Network**: The traffic data is collected through an irregular sensor network across major roads. Each sensor represents a region, and the region relationships are modeled using a graph $G = (V, E, A)$, where V are regions, E are edges, and $A \in \mathbb{R}^{R \times R}$ is a weighted adjacency matrix capturing spatial dependencies, based on geographical distance.
- ii) **Grid-based Traffic Network**: the urban area is partitioned into a regular grid of uniform square blocks, often with dimensions of 1km × 1km. This grid-like spatial representation allows the transportation network to be modeled using a graph structure, where each grid block is represented as a node, and the connections between neighboring regions are captured as edges.

Traffic Flow Forecasting. The task of traffic forecasting is to use data from the past H time steps ($\mathbf{X}_{t_{I-H+1}:t_I}$) to predict traffic data $\mathbf{Y}_{t_{I+1}:t_{I+F}}$ for the future F time steps. This involves exploring the complex spatial and temporal patterns inherent in traffic data to uncover the underlying dynamics that drive changes in traffic flow.

$$\mathbf{Y}_{t_{I+1}:t_{I+F}} = f(\mathbf{X}_{t_{I-H+1}:t_I}) \tag{1}$$

where $f(\cdot)$ represents the spatio-temporal prediction function.

Spatio-Temporal Generalization. In this work, we propose a foundation model to empower traffic prediction with strong generalization capabilities across both spatial and temporal dimensions.

(i) **Temporal Generalization for Long-term Forecasting**. Most existing traffic prediction models are limited to forecasting only short-term (*e.g.*, next hour) traffic variations. These models often struggle to generalize well to long-term traffic forecasting due to their poor temporal generalization ability. However, in practical applications, long-term traffic forecasting (*e.g.*, days or weeks into the future) is critically important and highly beneficial for various transportation planning and management tasks.

(ii) **Spatial Generalization for Zero-Shot Forecasting**. In real-world intelligent transportation systems, data heterogeneity and scarcity pose significant challenges. For instance, traffic flow patterns can vary substantially across different geographical regions within a city, and also differ considerably between cities. This presents a major obstacle for current traffic flow prediction models, as they often struggle to generalize their performance to new locations where little or no historical data is available.

Given the discussion above, we can formally define the traffic flow prediction task with strong spatio- temporal generalization:

$$\tilde{\mathbf{Y}}_{t_{I+1}:t_{I+F}} = \bar{f}(\tilde{\mathbf{X}}_{t_{I-H+1}:t_I}) \tag{2}$$

Different from the predictive function ($f(\cdot)$) in existing spatio-temporal models, the proposed $\bar{f}(\cdot)$ is a generalized spatio-temporal foundation model. This model can be directly utilized to make predictions on downstream unseen traffic data $\tilde{\mathbf{X}}$, which has no overlap with the training data $\mathbf{X}$. $\bar{f}(\cdot)$ is pre-trained on a diverse set of traffic data sources. It captures the underlying patterns and relationships that govern traffic dynamics across different spatial and temporal contexts. This enables zero-shot traffic flow forecasting without requiring any additional fine-tuning or adaptation steps.

3 METHODOLOGY

3.1 Spatio-Temporal Embedding for Distribution Shift Generalization

The design of our spatio-temporal embedding layer is driven by the need to address the distribution shift across both spatial and temporal dimensions in traffic prediction tasks. In real-world intelligent transportation systems, traffic flow patterns can vary considerably across different geographical regions and time periods. This inherent data heterogeneity presents a major obstacle for traffic prediction models, as they need to generalize their performance to new environments or scenarios where the underlying data distribution may be drastically different from the training data.

3.1.1 **In-Context Normalization for Zero-Shot Generalization.** Existing approaches often leverage statistical features of the training data, such as mean and standard deviation, for data normalization. However, these summary statistics may be inadequate or non-transferable when the test data exhibits significant data heterogeneity and has no spatial overlap with the training data distribution. To address this challenge and accommodate zero-shot traffic prediction tasks, we employ instance normalization $\mathbf{IN}(\cdot)$ to process the data. This approach utilizes the mean and standard deviation of the individual input instances $\mathbf{X}_r \in \mathbb{R}^T$ for each region, rather than relying on the global training set statistics. The formalization of this process is as follows:

$$\bar{\mathbf{X}}_{r,t} = \mathbf{IN}(\mathbf{X}_{r,t}) = \frac{\mathbf{X}_{r,t} - \mu_r}{\sigma_r} \tag{3}$$

Where $\mu(\mathbf{X}_r)$ and $\sigma(\mathbf{X}_r)$ are the mean and standard deviation of the input instance $\mathbf{X}_r$, respectively. Subsequent denormalization is applied to the prediction outputs to achieve more accurate results. Studies [10, 34] indicate that instance normalization effectively mitigates distribution shifts between training and test sets, advantageous for our zero-shot traffic prediction scenarios.

3.1.2 **Patch Embedding for Efficient Long-Term Prediction.** The proposed OpenCity model is designed to address long-term traffic prediction, which often involve processing an increased number of input time steps. This can lead to significant computational and memory overhead. To mitigate these issues, we employ a patch-based approach [5, 27] to partition the data along the temporal dimension. We define P as the patch length, specifying the number of time steps grouped into a single patch, and S as the stride size, determining the overlap between successive patches. After the patch operation, the input data $\bar{\mathbf{X}}_r \in \mathbb{R}^T$ is reshaped to $\mathbf{X}_r^P \in \mathbb{R}^{P\times N}$, where N is the number of blocks, calculated as $N = \frac{T-P}{S} + 1$. Our patch embedding scheme offers several key advantages:

- **Handling Temporal Distribution Shift**. By considering one hour of traffic data as the length of a single patch and adjusting the stride accordingly ($S = P$), our model can effectively handle the temporal distribution shifts often encountered in long-term traffic prediction. This allows the model to capture and adapt to the evolving patterns in traffic data over extended time horizons.
- **Computational Efficiency**. The patch-based processing significantly reduces the computational and memory requirements, as the model operates on a smaller number of input patches rather than the entire temporal sequence. This enables more efficient and scalable long-term traffic prediction, making the model suitable for real-world deployments.

After the patch-based partitioning of the input data, we employ a linear transformation and positional encoding to obtain the final spatio-temporal data embedding $\mathbf{E}_r \in \mathbb{R}^{P\times d}$. This embedding serves as the input to the subsequent model components:

$$\mathbf{E}_r = W_e \cdot \mathbf{X}_r^P + \mathbf{PE}(\mathbf{X}_r^P) \tag{4}$$

Here, $W_e \in \mathbb{R}^{N\times d}$ is the weight parameter, and **PE** denotes the Sinusoidal Position Encoding [35], which helps the model capture temporal relationships within each patch. Our temporal patch embedding schema compresses the input time steps, enabling the model to efficiently perform long-term traffic prediction and overcome challenges posed by increased input time steps, which is crucial for accurate long-term forecasting.

3.2 Spatio-Temporal Context Encoding

To capture the complex spatio-temporal patterns inherent in traffic data, our model integrates both temporal and spatial context cues. By explicitly modeling the interplay between these two critical

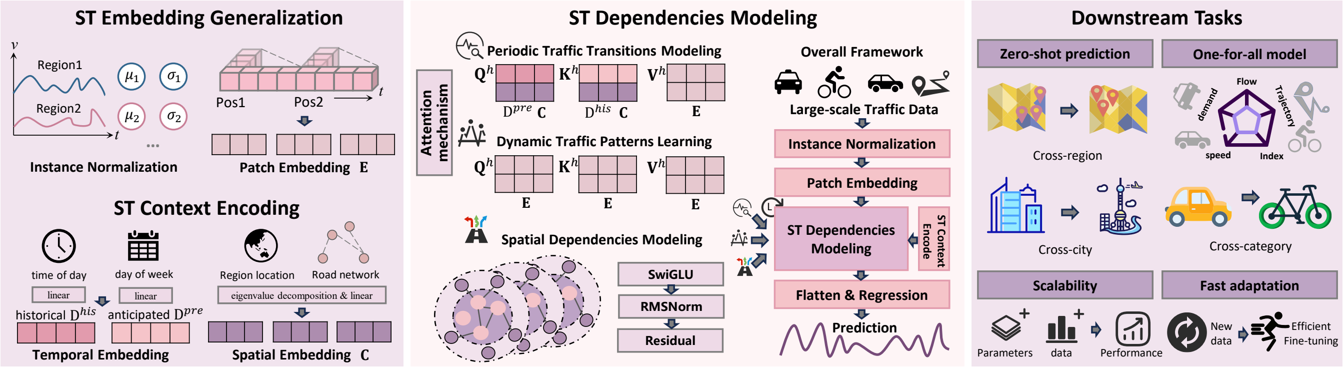

Figure 2: The overall architecture of Spatio-Temporal Foundation Model OpenCity.

dimensions, OpenCity is able to better understand the multifaceted factors influencing transportation patterns. This integrated approach enables our framework to generate more accurate forecasts across a diverse range of time horizons and geographic areas.

3.2.1 **Temporal Context Encoding.** Our model effectively captures the distinct temporal patterns in traffic data, such as periodic variations caused by daily or weekly routines, as well as complex non-linear dependencies over longer timescales. Specifically, we leverage patch-based segmentation to extract features related to the time of day, $z^{(d)} \in \mathbb{R}^T$, and the day of the week, $z^{(w)} \in \mathbb{R}^T$, and employ linear layers to generate time-specific embeddings that encode these temporal context cues. By explicitly modeling the periodic nature of traffic flow, our approach is well-equipped to make accurate predictions, even for long-term forecasting horizons where these temporal patterns become increasingly relevant.

$$\mathbf{D} = \text{concat}[W_1 z_P^{(d)}, W_2 z_P^{(w)}] \tag{5}$$

The temporal context is extracted as $z_P^{(d)}, z_P^{(w)} \in \mathbb{R}^{P\times N}$ after patch-based segmentation. We use linear layers parameterized by $W_1, W_2 \in \mathbb{R}^{N\times d/2}$ to generate time-of-day and day-of-week embeddings, which are then concatenated into the final temporal encoding $\mathbf{D} \in \mathbb{R}^{P\times d}$ and integrated into our model.

3.2.2 **Spatial Context Encoding.** Traffic patterns vary across regions due to their unique geographical characteristics, such as the higher volumes experienced at transportation hubs. To capture these regional attributes, we incorporate the underlying spatial context within the traffic network. First, we calculate the normalized Laplacian matrix $\Delta = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{I}$ and $\mathbf{D}$ are the identity and degree matrices, respectively. We then perform an eigenvalue decomposition, obtaining $\Delta = \mathbf{U}\Lambda\mathbf{U}^{\mathrm{T}}$, with $\mathbf{U}$ and Λ as the eigenvalue and eigenvector matrices. The k smallest nontrivial eigenvectors are used as the region embeddings $\Phi \in \mathbb{R}^{R\times k}$, encoding the structural information of the traffic network. These embeddings are then processed through a linear layer to obtain the final spatial encodings $\mathbf{C} \in \mathbb{R}^{R\times d}$.

3.3 Spatio-Temporal Dependencies Modeling

3.3.1 **Temporal Dependencies Modeling.** To highlight the strong generalization capabilities of our foundation model for long-term forecasting, it can effectively handle both periodic and dynamic traffic patterns. Specifically, we build our pre-trained spatio-temporal model upon our developed TimeShift Transformer architecture to encode time-dependent relationships. This allows our model to capture the traffic patterns from two complementary perspectives:

- **Periodic Traffic Transitions**. Our model captures the periodic, recurring traffic patterns, such as hourly, daily, and weekly cycles. By encoding these cyclical variations, our approach can better account for the inherent regularities in transportation network.
- **Dynamic Traffic Patterns**. Beyond just periodic variations, our temporal encoder also captures the complex, non-linear temporal dynamics and evolving trends in the traffic data over time.

Our model's ability to encode both the periodic and dynamic aspects of temporal information enables strong generalization capabilities, making it well-suited for real-world traffic forecasting across diverse time horizons. This versatile understanding of the multifaceted factors influencing transportation patterns allows accurate predictions of not only regular, recurring traffic behaviors, but also unpredictable, evolving changes in conditions.

Modeling Periodic Traffic Transitions. To learn the periodic patterns in the traffic, we leverage both temporal embeddings $\mathbf{D}$ and spatial embeddings $\mathbf{C}$. Our objective is to uncover the correlations between historical traffic patterns and future instances. Specifically, we update the temporal embeddings into two components:

- $\mathbf{D}^{his} \in \mathbb{R}^{R\times P\times d}$: This captures the historical temporal signals.
- $\mathbf{D}^{pre} \in \mathbb{R}^{R\times P\times d}$: This represents the anticipated temporal information for future predictions.

Our model explicitly models both historical and future-oriented temporal patterns, allowing it to better learn and leverage the periodic nature of the traffic time series.

To ensure dimension alignment, we extend the temporal embeddings along the spatial domain, and the spatial embeddings across the temporal axis. We then construct the inputs to a multi-head attention mechanism: the future spatio-temporal embeddings as the query, the historical spatio-temporal embeddings as the key, and the historical spatio-temporal data representations as the value.

$$Q_r^h = W_q^h(\mathbf{D}_r^{pre} + \bar{\mathbf{C}}_r); \quad K_r^h = W_k^h(\mathbf{D}_r^{his} + \bar{\mathbf{C}}_r); \quad V_r^h = W_k^h\mathbf{E}_r \tag{6}$$

$$\mathbf{M}_r^h = Attention(Q_r^h, K_r^h, V_r^h) = \text{softmax}(\frac{\delta_a(Q_r^h K_r^{h^\top})}{\sqrt{d_h}})V_r^h \tag{7}$$

Here, W_q^h, W_k^h, and W_v^h are learnable weight parameters of size $\mathbb{R}^{d\times d_h}$, and δ_a is a dropout operation. $\bar{\mathbf{C}}_r \in \mathbb{R}^{P\times d}$ is the updated spatial embedding, and $\mathbf{M}_r^h \in \mathbb{R}^{P\times d_h}$ is the output of the periodic association encoding from the h-th attention head. We incorporate RMSNorm [45] to improve training stability, explicitly modeling the relationship between historical and future temporal information to enable the discovery of periodic spatio-temporal traffic patterns.

Learning with Dynamic Traffic Patterns. This module designed to capture the dynamic dependencies among different time segments. For instance, events such as sudden traffic accidents may lead to a sharp decrease in traffic speed and traffic volume. Thus, we empower OpenCity to model these types of dynamic relationships, which are crucial for understanding and predicting traffic patterns. To achieve this, we employ an attention mechanism similar to the periodic traffic transition encoding. The difference is that the query (Q), key (K), and value (V) inputs are replaced by the normalized output (M) from the previous step. This modification allows the model to focus on learning the dynamic dependencies between different time segments, rather than just the periodic patterns. The resulting temporal representation $\mathbf{H} \in \mathbb{R}^{R\times P\times d}$ captures these dynamic spatio-temporal relationships.

3.3.2 **Spatial Dependencies Modeling.** Capturing spatial dependencies is a crucial aspect of our model design, as transportation networks exhibit strong spatial correlations where the traffic conditions in one region are often influenced by the states of neighboring areas. To model these spatial associations, we employ graph convolutional networks (GCNs). The formalization is as follows:

$$\mathbf{G}_t = \delta[\alpha\mathbf{H}_t + (1-\alpha)(W_g\bar{\mathbf{A}}\mathbf{H}_t)] \tag{8}$$

Here, $\bar{\mathbf{A}}$ is the normalized adjacency matrix weights, and $W_g \in \mathbb{R}^{d\times d}$ are learnable parameters. The α parameter balances original information preservation, and the δ function helps mitigate overfitting. For subsequent computations, we use residual connections, RMSNorm (RN), and the SwiGLU [31] activation, which has been successful in large language models.

$$\mathbf{O}_{r,t}^{(l)} = \text{SwiGLU}[\text{RN}(\mathbf{G}_{r,t}^{(l)} + \mathbf{O}_{r,t}^{(l-1)})] + \mathbf{G}_{r,t}^{(l)} \tag{9}$$

$$\text{SwiGLU}(\mathbf{E}_{r,t}) = W_c[\sigma(W_a\mathbf{E}_{r,t}) \cdot W_b\mathbf{E}_{r,t}] \tag{10}$$

Here, $\mathbf{O}_{r,t}^{(l)} \in \mathbb{R}^d$ represents the final output of the l-th layer spatio-temporal encoding network, σ is the Swish activation function, and $W_a, W_b, W_c \in \mathbb{R}^{d\times d}$ are trainable parameters. The proposed model captures complex spatio-temporal dependencies by stacking multiple layers of this spatio-temporal encoding network, allowing it to learn the intricate relationships within transportation networks.

4 EVALUATION

In our experimental evaluation, we address the following six key research questions to validate the capabilities of our model:

- **RQ1**: Can the proposed OpenCity model effectively generalize to zero-shot and long-term traffic forecasting scenarios?
- **RQ2**: How effectively does the proposed OpenCity framework integrate cross-data traffic patterns into a single and coherent model in supervised learning environments?
- **RQ3**: Does the model have the capability to rapidly generalize to new transportation forecasting demands as they emerge?
- **RQ4**: What are the individual contributions of the different modules to the performance gains of the OpenCity approach?
- **RQ5**: How the model's parameter count and training data volume scale to affect its overall forecasting performance?
- **RQ6**: Compared to existing large-scale spatio-temporal prediction models, what key advantages does our approach offer in terms of prediction performance and efficiency?

4.1 Experimental Setup

4.1.1 **Data Sources and Characteristics.** The model's generalization capabilities and predictive performance were extensively evaluated using a diverse set of large-scale, real-world public datasets covering various traffic-related data categories, including **Traffic Flow**, **Taxi Demand**, **Bicycle Trajectories**, **Traffic Speed Statistics**, and **Traffic Index Statistics**, from regions across the United States and China, such as New York City, Chicago, Los Angeles, the Bay Area, Shanghai, Shenzhen, and Chengdu. In addition, we released 3 versions OpenCity based on parameter size: OpenCity$_{\text{mini}}$ (2M), OpenCity$_{\text{base}}$ (5M), and OpenCity$_{\text{plus}}$ (26M). Further details on all experimental datasets are provided in the Appendix.

In the pre-training stage of the OpenCity approach, we leveraged a wealth of transportation data, encompassing the traffic flow, traffic speed, and taxi demand datasets. This expansive dataset covers a total of 10,110 regions and 352,796 time points, amounting to an astounding 151,089,924 observations. For the testing phase, we selected data that was outside the training set, allowing us to assess the model's generalization performance across a diverse range of traffic prediction scenarios. These scenarios include:

- **Cross-Region Zero-Shot Evaluation**: Assessing the OpenCity's ability and robustness to generalize to unseen regions within a city, without the need for additional training.
- **Cross-City Zero-Shot Evaluation**: Examining the model's capacity to adapt to completely new cities, leveraging the knowledge acquired during the original training process.
- **Cross-Task Zero-Shot Evaluation**: Testing the model's ability to forecast different types of traffic-related data, including traffic flow, speed, and demand, without any additional training.
- **Unified Model Supervised Evaluation**: Evaluating the versatility and adaptability of our unified OpenCity within a supervised learning framework, focusing on its ability to handle diverse spatio-temporal traffic scenarios.
- **Cross-Data Fast Adaptation Evaluation**: Measuring the model's cost-efficient adaptation capabilities to new traffic datasets while requiring only a small amount of additional training data, in contrast to the need for extensive retraining from scratch.

Table 1: Comparison of performance between OpenCity in zero-shot setting and the baselines in full-shot setting.

Dataset	Evaluation	Zero-shot	Full-shot									
		OpenCity$_{\text{plus}}$	PDFormer	STWA	MSDR	AGCRN	MTGNN	STSGCN	GWN	ASTGNN	STGCN	TGCN
CAD3	MAE	**15.88**	20.28	21.65	22.74	18.72	17.59	21.88	16.94	23.60	20.24	19.56
	RMSE	**27.03**	36.43	37.55	37.15	31.93	28.92	34.52	28.81	39.35	34.34	30.82
	MAPE	**21.94%**	25.19%	26.85%	32.37%	25.45%	25.22%	30.20%	22.98%	41.22%	25.33%	28.15%
CAD5	MAE	11.09	12.89	14.43	15.44	12.88	11.70	13.87	**10.69**	12.58	13.76	13.07
	RMSE	**18.96**	21.18	24.14	29.52	23.78	20.30	22.32	19.75	21.23	27.49	21.56
	MAPE	27.90%	28.82%	29.34%	32.49%	26.29%	26.75%	32.84%	**25.98%**	30.56%	32.89%	32.65%
PEMS07M	MAE	4.50	4.62	4.54	4.81	4.61	4.52	4.56	**4.17**	4.39	4.44	4.88
	RMSE	8.21	8.36	8.57	8.61	8.63	8.06	8.05	**7.84**	8.21	8.37	8.38
	MAPE	12.20%	13.74%	12.91%	13.90%	12.81%	13.11%	12.70%	**11.46%**	12.58%	12.59%	13.94%
TrafficSH	MAE	**0.55**	0.77	1.42	1.84	1.33	0.81	1.66	0.76	0.69	1.60	1.79
	RMSE	**0.85**	1.23	2.49	3.55	2.45	1.26	3.33	1.39	1.09	3.14	2.65
	MAPE	**8.01%**	8.34%	9.92%	9.91%	8.51%	8.29%	9.33%	9.23%	8.05%	8.04%	17.75%
CHI-TAXI	MAE	**1.91**	4.03	3.70	3.55	3.60	3.27	4.87	3.56	3.28	3.09	4.02
	RMSE	**4.42**	12.82	11.49	10.39	11.31	9.87	14.40	11.27	10.32	9.54	11.70
	MAPE	40.07%	44.42%	42.52%	52.88%	46.48%	**39.38%**	104.64%	41.31%	42.82%	42.47%	60.25%
NYC-BIKE	MAE	**6.32**	7.33	7.97	8.08	8.13	6.48	6.85	7.61	6.44	8.01	7.75
	RMSE	11.60	13.01	14.35	13.97	14.46	**11.49**	11.98	13.56	11.62	13.94	13.49
	MAPE	70.06%	65.44%	64.19%	77.56%	78.36%	61.52%	68.44%	**58.41%**	63.90%	63.02%	85.27%

4.2 Zero-shot vs. Full-shot Performance (RQ1)

In this section, we focus on evaluating the zero-shot generalization capability of the OpenCity. After its pre-training phase, the OpenCity is directly applied to zero-shot prediction tasks on various downstream datasets, including **Cross-Region**, **Cross-City**, and **Cross-Category** settings. In contrast, the baseline models are developed under a supervised learning framework, where they undergo training on these downstream datasets before being tested. The experimental setups are labeled as **Zero-shot** for the OpenCityand **Full-shot** for the baselines, with the results displayed in Table 2. The best performance is highlighted in bold, while the second-best performance is indicated with an underline.

(i) Outstanding Zero-shot Prediction Performance. OpenCity achieves significant zero-shot learning breakthroughs, outperforming most baselines even without fine-tuning. This highlights the approach's robustness and effectiveness at learning complex spatio-temporal patterns in large-scale traffic data, extracting universal insights applicable across downstream tasks.

OpenCity consistently secures top or second positions on several datasets, and maintains a competitive performance gap within 8% MAE even when not leading. This outstanding zero-shot prediction performance underscores the OpenCity's versatility and adaptability in handling diverse traffic datasets without extensive retraining. A crucial advantage is its readiness for immediate deployment in new scenarios, significantly reducing the time and resources typically required by traditional supervised approaches, offering substantial benefits for practical applications.

(ii) Exceptional Cross-Task Generalization. Our model was evaluated across four distinct traffic data categories: traffic flow (CAD3, CAD5), traffic speed (PEMS07M, TrafficSH), taxi demand (CHI-TAXI), and bicycle trajectories (NYC-BIKE). The baseline analysis revealed that while various models performed exceptionally well on specific data types, none could consistently deliver top-tier results across all categories. For instance, GWN, STGCN, and ASTGCN each exhibited remarkable capabilities in predicting traffic flow (CAD3, CAD5), taxi demand (CHI-TAXI), and traffic speed (TrafficSH) respectively. However, they struggled to maintain that level of performance in other domains. In contrast, the OpenCity consistently delivered high-quality results across all tested categories, underscoring its exceptional robustness and versatility.

To assess the versatility of our OpenCity framework, we evaluated its cross-category zero-shot generalization during testing. This involved incorporating bicycle trajectory data (NYC-BIKE), despite its absence in pretraining. The results were highly promising, as OpenCity maintained excellent performance on both MAE and RMSE metrics, further validating its universality and ability to adapt across diverse data types in real-life urban scenarios.

(iii) Strong Long-term Forecasting Capabilities. A key strength of our OpenCity architecture is its exceptional temporal generalization ability, which enables it to outperform baseline methods in long-term traffic prediction tasks. Many existing models often struggle to maintain accurate forecasts over extended time horizons, as they tend to overfit to historical patterns and fail to adequately capture the dynamic and evolving nature of traffic conditions. In contrast, OpenCity has demonstrated a remarkable capacity to learn universal spatio-temporal representations from diverse traffic data sources. This allows the model to generate robust predictions that remain reliable even as traffic patterns shift and evolve over time.

4.3 Exceptional Supervised Performance (RQ2)

To further validate the advantage of our OpenCity over current baseline methods, we conducted a supervised learning evaluation. In this setting, the OpenCity is directly compared to the baseline approaches, which are built in an end-to-end training manner. This supervised evaluation allows us to assess the OpenCity's performance when trained on the target datasets, in contrast to the zero-shot generalization setting previously explored.

As illustrated in Table 2, the results indicate that our OpenCity maintains excellent performance in the supervised setting and holds a leading advantage in most evaluation metrics. Additionally, we observed that most baseline models underperformed on the CAD-X dataset, possibly due to their tendency to overfit historical spatio-temporal patterns, making it challenging for them to generalize to

Table 2: Evaluation of predictive performance across different models in a supervised setting.

Model	Dataset	CAD8-1			CAD8-2			CAD12-2			PEMS-BAY			NYC-TAXI		
	Metrics	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
TGCN		27.68	45.09	20.18%	23.43	37.03	15.55%	36.53	60.01	61.60%	2.94	6.33	7.23%	6.10	12.70	80.39%
STGCN		31.26	49.91	24.26%	24.30	39.59	18.61%	34.60	62.47	52.49%	2.72	5.96	6.55%	4.17	9.19	45.54%
ASTGCN		29.38	46.84	22.51%	24.24	38.14	18.36%	35.19	57.87	59.73%	2.82	6.31	7.06%	5.45	13.44	59.05%
GWN		29.03	49.26	23.23%	25.17	41.47	18.50%	38.05	69.89	64.77%	2.66	5.60	5.94%	4.28	10.59	41.82%
STSGCN		32.38	53.28	25.63%	24.60	41.23	18.54%	37.00	63.19	58.29%	2.87	6.10	6.80%	5.03	10.96	65.17%
MTGNN		31.60	53.01	25.82%	21.85	34.02	14.20%	36.20	65.55	56.86%	2.59	5.43	**5.87%**	3.72	7.94	43.35%
AGCRN		34.98	58.37	26.82%	24.40	39.21	16.49%	39.91	74.47	63.36%	2.65	**5.32**	6.05%	4.55	10.11	52.45%
MSDR		28.99	46.91	23.60%	26.67	41.50	18.96%	40.88	72.87	69.47%	2.94	5.68	6.42%	4.22	9.08	55.50%
STWA		32.09	53.26	25.62%	26.57	44.01	20.21%	41.21	77.78	67.72%	2.74	5.65	6.03%	6.05	15.22	54.81%
PDFormer		29.64	49.79	20.65%	24.32	38.95	14.84%	34.23	59.70	50.92%	2.75	6.15	6.50%	3.64	8.24	37.40%
OpenCity$_{plus}$		**22.50**	**39.06**	**15.50%**	**17.95**	**29.61**	**10.97%**	**24.20**	**39.23**	**33.22%**	**2.59**	5.67	5.99%	**3.10**	**6.45**	**37.39%**

Table 3: Evaluation of OpenCity's Fast Adaptability.

Model	CD-DIDI			SZ-DIDI		
	MAE	RMSE	Cost	MAE	RMSE	Cost
STGCN	3.29	4.68	38.7	2.87	4.07	46.8
GWN	3.33	4.75	38.8	3.14	4.47	47.8
MTGNN	3.38	4.80	32.38	3.08	4.32	40.4
AGCRN	3.55	5.03	59.4	3.16	4.54	72.0
STWA	3.29	4.74	115.27	3.36	4.74	145.4
PDFormer	3.00	4.31	73.9	2.75	4.06	209.1
			Zero-shot			
OpenCity$_{mini}$	5.05	6.81	-	3.30	4.62	-
OpenCity$_{base}$	5.60	8.88	-	3.71	6.15	-
OpenCity$_{plus}$	6.03	9.50	-	3.68	5.58	-
			Efficient Fine-tuning			
OpenCity$_{mini}$	3.03	4.36	**2.4**	2.42	3.62	**2.8**
OpenCity$_{base}$	2.99	4.30	3.3	2.40	3.59	4.1
OpenCity$_{plus}$	**2.97**	**4.29**	12.3	**2.36**	**3.55**	14.6

long-term traffic dependency modeling. In contrast, the OpenCity architecture effectively extracts universal periodic and dynamic spatio-temporal representations from our pre-training stage, addressing the issue of poor prediction performance caused by cross-time and cross-location distribution shift.

4.4 Model Fast Adaptation Capabilities (RQ3)

This section evaluates the swift adaptation abilities of our OpenCity for downstream tasks. We focus on a previously unseen traffic dataset and employ an "Efficient Fine-tuning" approach, where only the model's prediction head (the last linear layer) is updated with a maximum of three training epochs. As detailed in Table 3, the zero-shot performance of OpenCity on some indicators is not as robust as the full-shot performance of the baseline models, likely due to variations in traffic patterns and data sampling. However, after efficient fine-tuning, OpenCity's performance substantially improves, outperforming all compared models. Remarkably, the training time for OpenCity comprises only 2%-32% of that required by the baselines. This rapid adaptability underscores OpenCity's potential as a foundational traffic forecasting model, capable of quickly adapting to new spatio-temporal data categories.

4.5 Ablation Study (RQ4)

To assess the individual contributions of the various components within our proposed OpenCity model, we conducted an ablation study using the CAD3 and NYC-BIKE datasets in zero-shot scenarios. The evaluation results are presented in Figure 3.

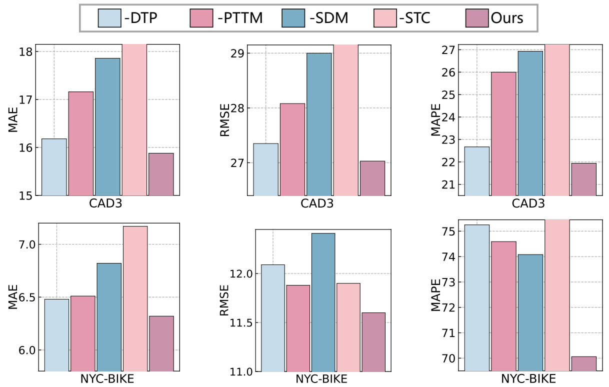

Figure 3: Ablation study of our OpenCity framework.

- **Impact of Learning from Dynamic Traffic Patterns**. In the *-DTP* variant, we removed the Dynamic Traffic Pattern modeling module, which resulted in a decrease in performance. By integrating this encoding process, the model is equipped to thoroughly analyze recent traffic patterns and adapt its predictions effectively in response to sudden changes in traffic conditions.
- **Impact of Periodic Traffic Transition Modeling**. In the *-PTTM* variant, we excluded the periodic correlation encoding, integrating temporal and spatial context directly into the spatio-temporal embedding. By modeling the mapping of traffic flow across historical-future time pairs, OpenCity captures the general laws governing the evolution of spatio-temporal patterns.
- **Impact of the Spatial Dependencies Modeling.** In the *-SDM* variant, we left the spatial encoding module unexplored. The analysis reveals that learning spatial relationships significantly enhances spatio-temporal prediction. By aggregating traffic information from dependent spatial regions, the model effectively captures dynamic traffic flow patterns, thereby providing a valuable auxiliary signal for zero-shot traffic prediction.
- **Impact of Spatio-temporal Context Encoding.** In the *-STC* variant, the encoding of temporal and spatial context information was omitted, resulting in a notable degradation in performance. Temporal context information helps the model identify and learn from common traffic patterns within specific periods, while regional embeddings encapsulate vital area-specific characteristics. These elements collectively provide valuable insights into understanding the dynamic spatio-temporal patterns across cities.

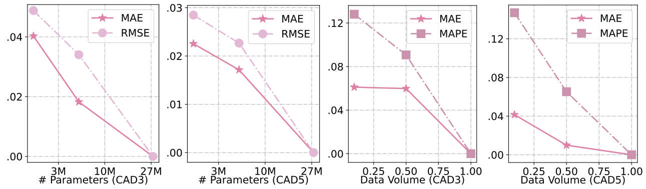

Figure 4: Exploration of OpenCity's scalability.

4.6 Scaling Law Investigation (RQ5)

We explored the scalability of OpenCity along both data and parameter dimensions, as illustrated in Figure 4. *Parameter Scalability*: We investigated 3 versions - OpenCity$_{\text{mini}}$ (2M), OpenCity$_{\text{base}}$ (5M), and OpenCity$_{\text{plus}}$ (26M) parameters. *Data Scalability*: For model$_{plus}$, we utilized 10%, 50%, and 100% of the pre-training data to explore the benefits of incorporating more data. To standardize comparisons, the vertical axis represents relative prediction error values. The results show that OpenCity's zero-shot generalization performance progressively improves as both parameter and data scale increase. This suggests OpenCity's ability to extract valuable knowledge from extensive datasets, with its learning capabilities enhanced by parameter expansion. The demonstrated scalability potential supports OpenCity's prospect as a foundational model for general transportation applications.

4.7 Comparison with Large Spatio-Temporal Pre-trained Models (RQ6)

In this section, we compare our proposed OpenCity with other prominent large spatio-temporal pre-trained models, including UniST [44] and UrbanGPT [20], known for their strong zero-shot generalization. We utilized the CHI-TAXI dataset, which was not included in the pre-training phase for any of the three models, for our evaluation. Detailed settings are provided in the Appendix. The results in Table 4 show that OpenCity maintains a significant performance advantage over other advanced large-scale spatio-temporal models. Additionally, OpenCity and UniST exhibit notable efficiency improvements compared to UrbanGPT. This may be due to UrbanGPT's reliance on the LLM for making predictions through a question-and-answer format, which hinders its ability to process batch data efficiently. The proposed OpenCity model achieves a win-win in performance and efficiency, highlighting its potential as a powerful large-scale model for traffic benchmarks.

Table 4: Comparison of large-scale ST pre-trained models.

Model	CHI-TAXI		
	MAE	RMSE	Cost (seconds)
UniST	2.94	7.88	1.3
UrbanGPT	3.26	7.10	4.5×10^4
OpenCity$_{\text{mini}}$	**1.74**	**3.80**	1.5

5 RELATED WORK

Deep Urban Traffic Prediction Models. The rise of deep learning has propelled deep spatio-temporal (ST) models to the forefront of the urban computing, known for their exceptional performance. These models analyze temporal and spatial correlations within historical data to forecast future trends. Prevalent temporal frameworks include recurrent neural networks (RNN)[43], temporal convolutional networks (TCN)[38], and attention networks [49]. Spatial correlations are typically encoded using graph convolution (GCN)[42], graph attention (GAT)[49], and hypergraph neural networks (HGNN) [39]. Ongoing research refines these temporal and spatial encoding strategies, including multi-scale [36] and multi-granularity learning [7], as well as adaptive [1, 37] and dynamic graph modeling [9, 48]. Researchers also explore cutting-edge techniques, such as neural ordinary differential equation for continuous time modeling [6, 14] for advancing the spatio-temporal learning.

Spatio-Temporal Self-Supervised Learning. Self-supervised learning (SSL) has emerged as an effective augmentation strategy for spatio-temporal learning. Existing SSL paradigms can be broadly categorized into: (1) Spatio-Temporal Contrastive SSL, which leverages contrastive learning to capture spatial and temporal correlations [18, 46]; (2) Spatio-Temporal Generative SSL, utilizing generative pretext tasks to model the underlying spatio-temporal dynamics via masked autoencoding [21, 30]; and (3) Spatio-Temporal Predictive SSL, incorporating auxiliary tasks that predict future spatio-temporal patterns with heterogeneity-aware data augmentation [11]. While these approaches aim to enhance the performance and generalization of spatio-temporal learning models, they are still limited by the constraints of zero-shot forecasting capabilities.

Leveraging Large Language Models in Urban Computing. The emergence of large language models (LLMs) has prompted research integrating their benefits into urban computing through three main paradigms: (1) Works like STLLM [23] and TPLLM [29] employ LLM architectures as ST encoders, fine-tuning for ST representation learning. (2) These efforts leverage LLMs' linguistic capabilities to enhance predictions, using in-context learning for model generalization enhancement [20] and interpretable generation [8]. (3) LLMs as ST Agents: This approach bridges LLMs with tools to facilitate traffic-related queries and reasoning based on human instructions [4]. While LLMs can improve spatio-temporal model performance, two key limitations exist: *First*, Computational Demands: LLMs' high computational requirements reduce efficiency, posing challenges for real-world deployment. *Second*, Reliance on POI Data: Most approaches heavily rely on the manually-collected, rich textual Point-of-Interest (POI) information, limiting generalization. To address these limitations, there is a need to develop more efficient and highly-generalized models that can learn universal spatio-temporal patterns across diverse transportation scenarios.

6 CONCLUSION

The work introduces OpenCity, a scalable spatio-temporal foundation model for traffic prediction that achieves precise zero-shot prediction performance across multiple traffic forecasting scenarios. By employing the Transformer encoder architecture as the backbone to model dynamic spatio-temporal dependencies and pre-training on the large-scale traffic datasets, OpenCity demonstrates exceptional zero-shot predictive performance on various downstream tasks, matching the results of state-of-the-art baseline models in full-shot scenarios. The proposed OpenCity framework can effectively handle data with varying distributions and boasts high computational efficiency. Along with the promising scaling laws observed, this paves the way for the development of a powerful, generalized traffic prediction solution that can be readily applied to diverse urban environments and transportation networks.

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] 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.

[3] 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.

[4] Longchao Da, Kuanru Liou, Tiejin Chen, Xuesong Zhou, Xiangyong Luo, Yezhou Yang, and Hua Wei. 2024. Open-ti: Open traffic intelligence with augmented language model. *International Journal of Machine Learning and Cybernetics* (2024), 1–26.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

[6] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. 2021. Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting. In *KDD*. 364–373.

[7] 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.

[8] Xusen Guo, Qiming Zhang, Junyue Jiang, Mingxing Peng, Hao, Yang, and Meixin Zhu. 2024. Towards Responsible and Reliable Traffic Flow Prediction with Large Language Models. arXiv:2404.02937

[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] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr DollÃ¡r, and Ross Girshick. 2022. Masked Autoencoders Are Scalable Vision Learners. In *CVPR*. 15979–15988.

[11] Jiahao Ji, Jingyuan Wang, Chao Huang, Junjie Wu, Boren Xu, Zhenhe Wu, Junbo Zhang, and Yu Zheng. 2023. Spatio-temporal self-supervised learning for traffic flow prediction. In *AAAI*, Vol. 37. 4356–4364.

[12] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. 2023. PDFormer: Propagation Delay-Aware Dynamic Long-Range Transformer for Traffic Flow Prediction. (2023), 4365–4373.

[13] Guangyin Jin, Yuxuan Liang, Yuchen Fang, Zezhi Shao, Jincai Huang, Junbo Zhang, and Yu Zheng. 2023. Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *Transactions on Knowledge and Data Engineering (TKDE)* (2023).

[14] Ming Jin, Yu Zheng, Yuan-Fang Li, Siheng Chen, Bin Yang, and Shirui Pan. 2023. Multivariate Time Series Forecasting With Dynamic Graph Neural ODEs. *TKDE* 35, 9 (2023), 9168–9180.

[15] 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.

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *CVPR*. 4015–4026.

[17] Xiaoliang Lei, Hao Mei, Bin Shi, and Hua Wei. 2022. Modeling network-level traffic flow transitions on sparse data. In *KDD*. 835–845.

[18] Rongfan Li, Ting Zhong, Xinke Jiang, Goce Trajcevski, Jin Wu, and Fan Zhou. 2022. Mining spatio-temporal relations via self-paced graph contrastive learning. In *KDD*. 936–944.

[19] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In *ICLR*.

[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. arXiv:2403.00813

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

[22] Zhonghang Li, Lianghao Xia, Yong Xu, and Chao Huang. 2024. FlashST: A Simple and Universal Prompt-Tuning Framework for Traffic Prediction. In *ICML*. 28978–28988.

[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

[24] Dachuan Liu, Jin Wang, Shuo Shang, and Peng Han. 2022. MSDR: Multi-Step Dependency Relation Networks for Spatial Temporal Forecasting. In *KDD*. 1042–1050.

[25] Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. 2023. LargeST: A Benchmark Dataset for Large-Scale Traffic Forecasting. In *Advances in Neural Information Processing Systems*.

[26] 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.

[27] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *ICLR*.

[28] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.021555* (2022).

[29] 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

[30] 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.

[31] Noam Shazeer. 2020. GLU Variants Improve Transformer. arXiv:2002.05202

[32] 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.

[33] David Alexander Tedjopurnomo, Zhifeng Bao, Baihua Zheng, Farhana Murtaza Choudhury, and Alex Kai Qin. 2020. A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *Transactions on Knowledge and Data Engineering (TKDE)* 34, 4 (2020), 1544–1561.

[34] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2017. Instance Normalization: The Missing Ingredient for Fast Stylization. arXiv:1607.08022

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

[36] 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.

[37] 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.

[38] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph wavenet for deep spatial-temporal graph modeling. In *IJCAI*.

[39] Xiaodong Yan, Tengwei Song, Yifeng Jiao, Jianshan He, Jiaotuan Wang, Ruopeng Li, and Wei Chu. 2023. Spatio-temporal hypergraph learning for next POI recommendation. In *SIGIR*. 403–412.

[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] Xueyan Yin, Genze Wu, Jinze Wei, Yanming Shen, Heng Qi, and Baocai Yin. 2021. Deep learning on traffic prediction: Methods, analysis, and future directions. *Transactions on Intelligent Transportation Systems (TITS)* 23, 6 (2021), 4927–4943.

[42] Bing Yu, Haoteng Yin, et al. 2018. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *IJCAI*.

[43] 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.

[44] Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. 2024. UniST: A Prompt-Empowered Universal Model for Urban Spatio-Temporal Prediction. arXiv:2402.11838

[45] Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. In *NeurIPS*.

[46] Qianru Zhang, Chao Huang, Lianghao Xia, Zheng Wang, Zhonghang Li, and Siuming Yiu. 2023. Automated Spatio-Temporal Graph Contrastive Learning. In *WWW*. 295–305.

[47] 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. *Transactions on Intelligent Transportation Systems (TITS)* (2020), 3848–3858.

[48] Yusheng Zhao, Xiao Luo, Wei Ju, Chong Chen, Xian-Sheng Hua, and Ming Zhang. 2023. Dynamic Hypergraph Structure Learning for Traffic Flow Forecasting. ICDE.

[49] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. GMAN: A Graph Multi-Attention Network for Traffic Prediction. In *AAAI*. 1234–1241.

A APPENDIX

In the appendix, we have provided comprehensive details on the OpenCity model, including the training and optimization procedures, dataset information, experimental settings, hyperparameter configurations, baseline explanations, evaluation metrics, and descriptions of the deployment experiments. This supplementary material aims to offer readers a thorough understanding of the technical aspects and experimental setup behind our work.

A.1 Model Training and Optimization

To learn robust spatio-temporal representations, our model leverages multiple large-scale traffic datasets for pre-training. This approach allows the model to capture the inherent complexities and dependencies within transportation networks by leveraging diverse data sources. During the training process, we randomly complete one step of training using data from a specific dataset, with one training epoch covering all available training data. Our model adopts a supervised training paradigm to complete this pre-training stage.

After obtaining the output from the L-th layer of the spatio-temporal encoding network, we flatten the features and use a linear layer to obtain the prediction $\mathbf{Y} \in \mathbb{R}^{R \times T}$ Following previous research [1, 22], we use the absolute error as the loss function $\mathcal{L}$:

$$\mathcal{L} = \frac{1}{RT} \sum_{r=1}^{R} \sum_{t=1}^{T} |(\hat{\mathbf{Y}}_{r,t} - \mathbf{Y}_{r,t})| \qquad (11)$$

Our model is designed to capture the intricate spatial and temporal dependencies inherent in transportation networks by leveraging multi-dataset pre-training. This approach enables the model to learn robust spatio-temporal representations that can generalize well to a wide range of scenarios, including spatial generalization with zero-shot prediction and temporal generalization with long-term forecasting. By pre-training on diverse datasets, the model can extract meaningful features and patterns that are transferable across different spatial and temporal domains, enhancing its versatility and performance in real-world traffic forecasting tasks.

A.2 Comprehensive Experimental Settings

A.2.1 **Data Sources and Characteristics.** To comprehensively evaluate the generalization capabilities and predictive performance of the OpenCity approach, we conducted training and testing across a diverse set of large-scale public real-world datasets. These datasets cover multiple traffic-related data categories from various regions, including the United States and China, spanning several major cities such as New York City, Chicago, Shanghai, and others. **(i) Traffic flow Data** [25, 32]: CAD-X and PEMS-X from California; **(ii) Taxi Demand Data** [20]: X-TAXI from New York City and Chicago; **(iii) Bicycle Trajectory Data** [20]: X-BIKE from New York City; **(iv) Traffic Speed Statistics** [44]: Traffic-X from major cities in China; **(v) Traffic Speed Statistics** [19, 42]: METR-LA, PEMS-BAY, and PEMS07M from Los Angeles, the Bay Area, and California; **(vi) Traffic Index Statistics** [26]: X-DIDI from Shenzhen and Chengdu.

The taxi demand datasets, bicycle trajectory datasets, and traffic speed statistics datasets (Traffic-X) used in this study are grid-based in nature, while the remaining datasets are sensor-based. To distinguish between the specific districts or cities represented within each dataset, the “X” placeholder is utilized.

The detailed statistics for all datasets used in this paper is shown in Table 5. Considering that a large number of regions might lead to excessive GPUs memory overhead, we segmented datasets CAD4, CAD7, CAD8, and CAD12 according to the order of nodes. Specifically, the number of regions in CAD4-1, CAD4-2, CAD4-3, and CAD4-4 are 621, 610, 593, and 528; in CAD7-1, CAD7-2, and CAD7-3 are 666, 634, and 559; in CAD8-1 and CAD8-2 are 510 and 512; and in CAD12-1 and CAD12-2 are 453 and 500. Additionally, the last column of Table 5 indicates whether each dataset was utilized for pre-training in OpenCity. To evaluate the model’s performance in a supervised setting, we reserved a portion of the data from the datasets involved in the supervised experiments for evaluation.

A.2.2 **Hyperparameter Configuration.** We release three different versions of the OpenCity model, each with varying parameter counts: **OpenCity**$_{\text{mini}}$ (**2M** parameters), **OpenCity**$_{\text{base}}$ (**5M** parameters), and **OpenCity**$_{\text{plus}}$ (**26M** parameters). We scale the OpenCity architecture by increasing the dimensions of the hidden layers and the number of layers in the spatio-temporal encoder, as detailed in Table 1. For our long-term traffic forecasting focus, we set the historical and future time spans (H and F) to 1 day. This configuration is determined by the aggregation frequency - when the frequency is 5 minutes, $H = F = 288$. Furthermore, we set the patch length P and stride S to one-hour time spans, which correspond to a value of 12 when the aggregation frequency is 5 minutes.

The regional embedding dimensionality k is set to 8 and the balancing weight α is set to 0.05. The dropout ratios for the attention matrix δ_a and the spatio-temporal network δ are set to 0.3 and 0.1, respectively. We maximize the batch size settings based on the GPU memory usage across the different models. Additionally, the hyperparameters for all baseline configurations adhere to the settings provided in the original papers or the official released codes. All training and testing are conducted on a server equipped with 8 × NVIDIA A100-SXM4-40GB GPUs.

A.2.3 **Detailed Experimental Setup.** The detailed settings of different experiments and the division of datasets are as follow:

Zero-shot Evaluation Setup: In the zero-shot evaluation, the proposed OpenCity model is directly used for testing, while the baselines are trained and tested in a supervised setting. For the traffic flow datasets (CAD3, CAD5) and the traffic speed datasets (PEMS07M, TrafficSH), the split ratios for training, validation, and testing sets are 0.5, 0.1, and 0.4, respectively. For the taxi demand dataset (CHI-TAXI) and the bicycle trajectory dataset (NYC-BIKE), these ratios are 0.2, 0.2, and 0.6. All the above datasets are unavailable during the pre-training phase of OpenCity.

Supervised Evaluation Settings: In the supervised scenario, the pre-trained OpenCity model is directly employed for evaluation. Contrary to the zero-shot setting, all these datasets are included in the pre-training phase. For the CAD8-1, CAD8-2, and CAD12-2 datasets, the proportions for training, validation, and testing are set at 0.8, 0.1, and 0.1, respectively. For the PEMS-BAY dataset, these ratios are 0.5, 0.1, and 0.4. In the case of the NYC-TAXI dataset, all data from 2016 to 2020 are used as the training set, data from the first two months of 2021 serves as the validation set, and data from the remaining ten months of 2021 is used as the test set.

Table 5: Statistical information of the datasets.

Dataset	Category	Geoinformation	# Regions	Sample rate	Time span(Y/M/D)	Obs.	Traffic net	For pre-training
CAD3	Traffic flow	California, USA	480	5 minutes	2020/03/01-2020/04/30	8.4M	Sensor	No
CAD4	Traffic flow	Bay Area, USA	2352	5 minutes	2020/01/01-2020/03/15	50.8M	Sensor	Yes
CAD5	Traffic flow	California, USA	211	5 minutes	2020/03/01-2020/04/30	3.7M	Sensor	No
CAD7	Traffic flow	Los Angeles, USA	1859	5 minutes	2020/01/01-2020/03/15	40.2M	Sensor	Yes
CAD8	Traffic flow	Los Angeles, USA	1022	5 minutes	2020/01/01-2020/03/15	22.1M	Sensor	Yes
CAD12	Traffic flow	Los Angeles, USA	953	5 minutes	2020/01/01-2020/03/15	20.6M	Sensor	Yes
PEMS04	Traffic flow	California, USA	307	5 minutes	2018/01/01-2020/02/28	5.2M	Sensor	Yes
PEMS08	Traffic flow	California, USA	170	5 minutes	2016/07/01-2020/08/31	3.0M	Sensor	Yes
NYC-TAXI	Taxi demand	New York City, USA	263	30 minutes	2016/01/01-2021/12/31	27.7M	Grid	Yes
CHI-TAXI	Taxi demand	Chicago, USA	77	30 minutes	2021/01/01-2021/12/31	1.3M	Grid	No
NYC-BIKE	Bike trajectory	New York City, USA	540	30 minutes	2020/01/01-2020/12/31	9.5M	Grid	No
TrafficZZ	Traffic speed	Zhengzhou, China	676	30 minutes	2022/03/05-2022/04/05	0.9M	Grid	Yes
TrafficHZ	Traffic speed	Hangzhou, China	672	30 minutes	2022/03/05-2022/04/05	0.9M	Grid	Yes
TrafficCD	Traffic speed	Chengdu, China	728	30 minutes	2022/03/05-2022/04/05	1.0M	Grid	Yes
TrafficJN	Traffic speed	Jinan, China	576	30 minutes	2022/03/05-2022/04/05	0.8M	Grid	Yes
TrafficSH	Traffic speed	Shanghai, China	896	30 minutes	2022/03/05-2022/04/05	1.3M	Grid	No
METR-LA	Traffic speed	Los Angeles, USA	207	5 minutes	2012/03/01-2022/04/30	3.6M	Sensor	Yes
PEMS-BAY	Traffic speed	Bay Area, USA	325	5 minutes	2017/01/01-2017/04/30	11.2M	Sensor	Yes
PEMS07M	Traffic speed	California, USA	228	5 minutes	2017/05/01-2017/08/31	2.9M	Sensor	No
SZ-DIDI	Traffic index	Shenzhen, China	627	10 minutes	2018/01/01-2018/02/28	5.3M	Sensor	No
CD-DIDI	Traffic index	Chengdu, China	524	10 minutes	2018/01/01-2018/02/28	4.5M	Sensor	No

Fast adaptability Evaluation Setup: In this experiment, we tested the OpenCity model using data categories not included in its training data, specifically the traffic index data from CD-DIDI and SZ-DIDI. The division ratios for training, validation, and testing were set at 0.5, 0.1, and 0.4, respectively. We utilize the 'Efficient Fine-tuning' approach, in which only the prediction head of the model (the last linear layer) is updated for a maximum of 3 training epochs. For baseline comparisons, we set the maximum number of training epochs at 100 and use an early stopping criterion after 15 epochs, without freezing any parameters. To maintain consistent timing expense statistics, we set the batch size at 64 for all models (except for PDFormer due to its higher GPU memory overhead).

In addition, we utilize OpenCity$_{\text{plus}}$ as the backbone model for our *ablation experiments* to assess the effectiveness of various modules. For *comparisons with large-scale spatio-temporal pre-trained models*, we directly use the open-source large model from the source repository to conduct performance evaluations. Taking into account the configurations of UrbanGPT and UniST, we employ the last month of CHI-TAXI data as the test set and evaluate the prediction accuracy of the first six steps across different models.

A.2.4 Evaluation Metrics for Traffic Prediction.

The experiments utilize three widely used evaluation metrics in traffic forecasting and regression tasks: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). MAE measures the average absolute difference between the predicted values and true labels, RMSE captures the standard deviation of the prediction errors, and MAPE represents the average absolute percentage difference between the predicted values and true labels. Lower values for these metrics indicate better performance of the model, as they quantify the errors between the predicted values and true labels from different perspectives.

A.2.5 Baseline Description.

We selected 10 advanced spatio-temporal prediction models as baselines, all of which have demonstrated considerable success in traffic prediction tasks. Given that most of these models are primarily designed for short-term predictions, we have enhanced their long-term prediction capabilities by applying patch embedding, as introduced in Section 3.1.2. The selected baseline models fall into the following three categories:
RNN-Based Spatio-temporal Prediction Models:

- **AGCRN** [1]: This model incorporates learnable node embeddings into recurrent neural networks (RNNs), enabling it to capture the region-specific spatio-temporal evolution patterns.

Table 6: Deployment study of OpenCity model.

Model	$\textbf{OpenCity}_{\text{plus}}$				
	MAE	RMSE	Cost (s)	Obs.	# Regions
CHI-TAXI	1.67	3.53	2.00	3,936	77
NYC-BIKE	3.88	6.29	2.10	25,920	540
CAD3	30.68	48.50	2.08	138,240	480
TrafficSH	0.51	0.78	2.22	43,008	896

- **MSDR** [24]: This method enhances RNNs by addressing the issue of long-term information forgetting. It achieves this by preserving multi-step hidden states at each time unit, facilitating the modeling of both long and short-term temporal relationships.

Attention-Based Spatio-temporal Neural Networks:

- **ASTGCN** [7]: This model leverages an attention mechanism to develop multi-granular spatiotemporal representations.
- **STWA** [3]: In this method, time-varying and location-specific parameters are integrated into the attention mechanism to capture time-aware and region-aware spatio-temporal dependencies.
- **PDFormer** [12]: This method employs attention networks to learn ST representations separately in the temporal and spatial dimensions. Delay-aware modeling is incorporated into the spatial attention mechanism to model traffic mobility patterns.

RNN-Based Spatiotemporal Prediction Models:

- **TGCN** [47]: This model employs GNNs to capture spatial correlations and utilizes RNNs to model temporal dependencies.
- **STGCN** [42]: In this method, graph convolutional networks and gated temporal networks are utilized to model spatial dependencies and temporal dependencies, respectively.
- **GWN** [38]: This model uses a learnable graph structure and 1-D dilated convolution to capture spatial and temporal correlations.
- **MTGNN** [37]: This method employs a learnable graph neural network to model spatial dependencies, and the diffusion convolution network is used to learn multi-scale temporal correlations.
- **STSGCN** [32]: It introduces a spatio-temporal graph construction method to simultaneously learn temporal and spatial correlations.

A.3 Model Deployment

In this section, we examine the deployment-related challenges associated with OpenCity. Given that the datasets for all our experiments are derived from real-world scenarios and align with actual prediction environments, the performance of our model in deployment settings has been validated in the aforementioned experiments. Our focus here extends to assessing the efficiency of OpenCity in practical applications. In contrast to offline prediction, online evaluation typically involves processing a single batch of predictions within a specific timeframe, dictated by the regular intervals at which traffic data is updated and aggregated (*e.g.*, every 5 minutes, every 2 hours, 1 day). We deployed OpenCity on a single NVIDIA A100-SXM4-40GB GPU and evaluated its prediction speed across cities of varying sizes. The task involved predicting the next day's traffic trends based on the data from the previous day.

According to the results presented in Table 6, OpenCity is capable of delivering predictions in an exceptionally short period. Even in cities with numerous regions and fine-grained prediction requirements, OpenCity completes individual predictions in less than 3 seconds. This remarkable efficiency not only demonstrates the model's robustness but also reinforces OpenCity's potential to serve as a foundational model for traffic prediction.