Spatio-Temporal Pyramid-Based Multi-Scale Data Completion in Sparse Crowdsensing

Wenbin Liu, Hao Du, En Wang, *Member, IEEE*, Jiajian Lv, Weiting Liu, Bo Yang, and Jie Wu, *Fellow, IEEE*

Abstract—**Sparse Crowdsensing has emerged as a crucial and flexible method for collecting spatio-temporal data in various applications, such as traffic management, environmental monitoring, and disaster response. By recruiting users and utilizing their diverse mobile devices, this approach often results in data that is both sparse and multi-scale, complicating the data completion process. Although numerous data completion algorithms have been developed to address data sparsity, most assume that the collected data is of the same or similar scale, rendering them ineffective for multi-scale data. To overcome this limitation, in this paper, we propose a spatio-temporal pyramid-based multi-scale data completion framework in Sparse Crowdsensing. The basic idea is to leverage a pyramid structure to efficiently capture the complex interrelations between different scales. We first develop a Spatial-Temporal Pyramid Construction Module (ST-PC) to handle multi-scale inputs, and then propose a Spatial-Temporal Pyramid Attention Mechanism (ST-PAM) to capture multi-scale correlations while reducing computational complexity. Furthermore, our method incorporates cross-scale constraints to optimize completion performance. Extensive experiments on four real-world spatio-temporal datasets demonstrate the effectiveness of our framework in multi-scale data completion.**

Index Terms—**Sparse crowdsensing, spatio-temporal data completion, multi-scale, spatio-temporal pyramid attention.**

I. Introduction

With the rapid development of smart devices and wireless communication technologies [1], [2], Crowdsensing [3] has emerged as a powerful paradigm for data collection. This approach leverages the participation of users to gather spatio-temporal data in a cost-effective and scalable manner. Nowadays, Crowdsensing has been widely adopted in various domains such as traffic management [4], environmental monitoring [5], and disaster response [6]. Despite its advantages in flexibility and scalability, Crowdsensing also presents significant problems, such as high collection costs and heterogeneity in the sensing capabilities of user devices. As a result, the gathered data are often sparse and multi-scale, which making it challenging to apply the data directly in practical applications.

To address data sparsity and missingness, Sparse Crowdsensing [7], [8] has emerged as an effective approach. Various data completion algorithms, including those based on compressive sensing or low-rank matrix factorization [9], [10], [11], [12], [13], have been explored to mitigate these issues. Note that most of the existing works are based on the assumption that the collected data are of the same or similar scales. However, in real-world scenarios, due to the diversity of devices and the complexity of sensing environments, data collected by different users usually *vary in scales*. Consider a scenario requiring comprehensive traffic flow information during a large-scale sudden event, where the recruitment of users via a crowd sensing platform becomes essential. This approach, however, inherently yields sparse data due to the opportunistic and non-uniform nature of mobile sensing. Such data sparsity significantly complicates the accurate inference of a complete spatio-temporal view, making it challenging to capture evolving traffic patterns or predict future conditions effectively. Furthermore, due to the diversity of user devices, the collected data often exhibit multi-scale characteristics. For example, consider that traffic data is collected by different devices: a smartphone, a vehicle, and a drone. Obviously, a smartphone covers a smaller area, indicating a finer scale, while vehicles and drones cover progressively larger regions, indicating coarser scales.

As shown in Fig. 1, different scales are represented by varying grid sizes: smaller grids for finer scales and larger grids for coarser ones. There is a numerical relationship between scales: for example, coarser values such as air quality (e.g., PM2.5 or humidity) can be derived by averaging finer-scale values, while coarser traffic flow data may be the sum of finer-scale nodes. Furthermore, Fig. 2 further illustrates this by showing a general correspondence between coarse-scale aggregations and underlying fine-scale trends, highlighting inherent cross-scale correlations valuable for data completion. Traditional single-scale data completion models fail to capture these complex cross-scale

Received 5 November 2024; revised 7 August 2025; accepted 13 August 2025. Date of publication 14 August 2025; date of current version 3 December 2025. This work was supported in part by National Natural Science Foundation of China under Grant 62272193 and Grant 62472194, in part by Changchun City Major Science and Technology Project under Grant 2024WX05, in part by Doctoral Research Innovation Program, Department of Education of Jilin Province under Grant 2024KC117, and in part by Graduate Innovation Fund of Jilin University under Grant 2025CX212 and Grant 2025CX214. Recommended for acceptance by Z. Li. *(Corresponding author: En Wang.)*

Wenbin Liu, Hao Du, En Wang, Jiajian Lv, Weiting Liu, and Bo Yang are with the College of Computer Science and Technology, Jilin University, Changchun 130012, China, and also with the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China (e-mail: liuwenbin@jlu.edu.cn; duhao22@mails.jlu.edu.cn; wangen@jlu.edu.cn; jjlv24@mails.jlu.edu.cn; liuwt23@mails.jlu.edu.cn; ybo@jlu.edu.cn).

Jie Wu is with the China Telecom Cloud Computing Research Institute, Beijing 100088, China, and also with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: jiewu@temple.edu).

Digital Object Identifier 10.1109/TMC.2025.3599322

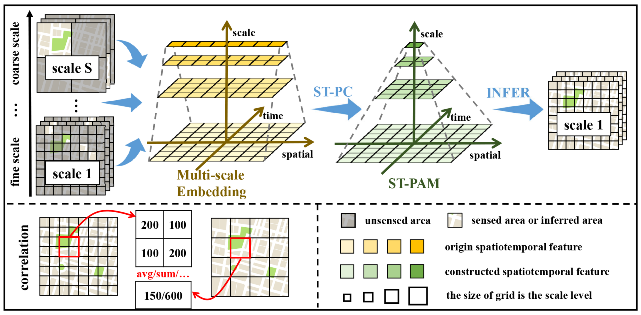


Fig. 1. The spatio-temporal pyramid-based multi-scale data completion.

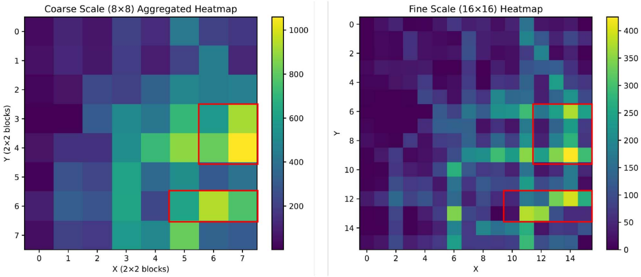


Fig. 2. Data-driven illustration of multi-scale data properties.

relationships, leading to incomplete representations and a bias towards specific scales, thus restricting a comprehensive understanding of the overall data structure. Therefore, addressing the *multi-scale data completion* in Sparse Crowdsensing is critical.

Some researchers have also acknowledged and explored the complexities associated with multi-scale data, such as in spatio-temporal data super-resolution [14], [15], [16], [17]. This approach aims to convert low-resolution data into high-resolution data, typically requiring the completion of low-resolution data before attempting super-resolution. However, the preliminary data completion step can introduce initial inaccuracies, leading to the accumulation of errors in the final results. Moreover, they primarily focus on the coarsest scale data, failing to fully leverage the intricate interconnections inherent in multi-scale spatio-temporal data. Similarly, some researchers have focused on exploring multi-scale correlations within time series data [18], [19], [20]. Their objective is to capture the inherent relationships between long-term and short-term time series, placing a significant emphasis on temporal multi-scale information. However, these studies frequently overlook spatial multi-scale connections and require high data completeness, limiting their use in environments with sparse data. In summary, these approaches still struggle with the challenges of *sparsity and complexity in multi-scale data*.

To this end, this paper aims to tackle the dual challenges of data sparsity and multi-scalability. Capturing multi-scale relationships is inherently complex. As shown in Fig. 2, while coarse and fine scales may reveal general trend correspondence, fine-scale units within a single coarse region often exhibit significant internal heterogeneity. In Sparse Crowdsensing, capturing these complex multi-scale correlations is further exacerbated by the limited and sparsely distributed data. Thus, how to *capture relationships* within sparse and multi-scale data is the first challenge. Note that the relationships within and across scales are highly complex. Simply computing the correlation between all data would significantly increase the input size and computational burden. Also, it could cause the model to overly focus on redundant information, thereby diminishing overall performance. Thus, how to reduce the *computational complexity* and focus on more useful information is the second challenge. Finally, to achieve better data completion performance, it is crucial to effectively utilize the sparse data collected at each scale and incorporate cross-scale correlations to enhance the model's capability. Thus, after capturing the relationships within and across scales, how to effectively *leverage sparse and multi-scale data* for data completion is the third challenge.

To address these challenges, this paper proposes a novel Spatio-Temporal Pyramid-Attention based Multi-scale Data Completion Framework in Sparse Crowdsensing. *To capture relationships within sparse and multi-scale data*, we introduce a Multi-Scale Embedding Layer to better represent features from diverse scales. Subsequently, we propose the Spatial-Temporal Pyramid Construction Module (ST-PC) to organize multi-scale inputs into a three-dimensional pyramid structure, providing a hierarchical representation conducive to relationship modeling. *To reduce the computational complexity and focus on more useful information*, we propose the Spatial-Temporal Pyramid Attention Mechanism (ST-PAM), using a pyramid structure for efficient correlation extraction within multi-scale data, while effectively reducing the computational complexity. Finally, *to effectively leverage sparse and multi-scale data*, we apply cross-scale constraint restrictions on the completion results.

Our work has the following contributions:

- We propose a novel framework specifically for multi-scale sparse data completion. This framework can effectively capture the correlations between multiple scales, and be used for inferring complete data at the finest scale. To the best of our knowledge, this is the first work to address the challenge of completing sparse data while considering multi-scale data collection scenarios.
- We develop the Spatial-Temporal Pyramid Construction Module (ST-PC). This module efficiently utilizes the characteristics of sparse multi-scale spatio-temporal data, and construct the multi-scale inputs as a three-dimensional pyramid structure.
- We propose the Spatial-Temporal Pyramid Attention (ST-PAM), a novel sparse attention mechanism to capture multi-scale data relationships. We show that, with optimal hyperparameters, ST-PAM achieves $O(1)$ maximum path length and $O(LT)$ computational complexity.
- Experimentally, our algorithm performs better compared to other baseline algorithms on four real-world datasets. With limited multi-scale data assistance, our approach significantly outperforms single-scale models.

The remainder of this paper is organized as follows. After reviewing the related works in Section II, we introduce the system model and formulate the problem in Section III. Then, the multi-scale completion methods are proposed in Sections IV, followed by the theoretical analysis in Section V. Finally, we evaluate the performance in Section VI and conclude this paper in Section VII.

II. Related Work

A. Data Completion in Sparse Crowdsensing

With the growing demand for fine-grained data and the high costs of data collection, Sparse Crowdsensing [7], [8] has emerged as an efficient and scalable solution for gathering data. In this paradigm, data completion plays a crucial role, as it leverages spatio-temporal correlations to infer missing data from the sparse samples provided by users.

Traditional methods focus on machine learning algorithms for data completion. For instance, Candès et al. [9] introduced the concept of compressed sensing, a method for signal reconstruction that can accurately recover signals from highly incomplete frequency information. Subsequently, Candès et al. [10] employed matrix completion, using the low-rank property of matrices to restore complete data. Alternatively, Wu et al. [11] proposed spatio-temporal kriging interpolation, aimed at addressing data sparsity problems.

With the advancements in deep learning, neural network-based methods have demonstrated significant potential in spatio-temporal data completion. For instance, Yuan et al. [21] proposed the STGAN model, employing generative adversarial networks (GANs) to effectively address data sparsity. Li et al. [22] utilized convolutional neural networks (CNNs) for spatio-temporal data completion, improving data resolution via super-resolution techniques. Furthermore, Wang et al. [23] introduced a Transformer-based completion-prediction framework for spatio-temporal data inference and long-term prediction. In another development, Wang et al. [24] focused on few-shot data completion in Sparse Crowdsensing. Although these approaches achieve notable success in data completion, they are primarily tailored for single-scale data, limiting their effectiveness in multi-scale environments.

B. Multi-Scale Model

Multi-scale models have been extensively researched in the field of Computer Vision. David et al. [25] proposed a theory on how pyramid structures process and understand images, offering key insights into the human visual system. Building on this, Zhang et al. [26] introduced a multi-scale feature pyramid, improving the ability to capture multi-scale correlations. With the development of the Transformer [27], many studies based on the Vision Transformer [28] have been carried out. For example, Ren et al. [29] proposed a Shunted Self-Attention mechanism, capable of capturing relationships between multiple scales within the single attention layer. Recently, Fan et al. [30] proposed Retentive Networks that integrate vision transformers with retentive memory to capture both local and global scale dependencies more effectively.

In urban management, the application of multi-scale data is significant. One application is super-resolution, which uses coarse-scale data to infer fine-scale data. Crivellari et al. [31] use GAN to up-scale urban settlements from satellite imagery. Zhou et al. [15] used mobile IoT data to infer fine-grained urban traffic and employed neural networks to solve ordinary differential equations; Zhang et al. [16] conducted research on spatio-temporal super-resolution of precipitation using GAN models, achieving notable results. Beyond super-resolution technology, multi-scale data has also been widely applied in time series data, to understand the relationships between long-term and short-term cycles. Wang et al. [20] introduced TimeMixer, employing a decomposable multi-scale mixing mechanism to integrate information across both fine and coarse scales. Liu et al. [18] reduced computational complexity while maintaining model accuracy and efficiency through a pyramid-structured attention mechanism. However, the multi-scale work on spatio-temporal data is difficult to apply to sparse data. Given the great success of pyramids in multi-scale works, we propose to apply the pyramid structure to multi-scale sparse data completion.

III. System Model and Problem Formulation

A. System Model

In this study, we explore a comprehensive spatio-temporal data application scenario. Users employ sensors of different scales to collect data from specific areas during various time periods. Unlike traditional data-completion frameworks, we are faced with multi-scale data scenarios. The core objective is to reconstruct complete datasets from these sparse spatio-temporal observations. To better illustrate this scenario, we have provided a clear definition of its key components.

Sense Map: To clearly illustrate the data collection process, we begin by defining the concept of the sense map. Specifically, the spatial map is partitioned into L discrete subareas at the finest scale to ensure precise spatial coverage. Furthermore, the entire data collection process is structured by dividing it into T equal-length time periods. During each time period, users are assigned to their respective subareas to collect data.

Scale: We categorize the sensing capabilities of sensors into S distinct levels, where a higher level indicates a coarser scale. Coarser sensors encompass the sensing areas of multiple finer sensors, covering an area C-times larger than the finer sensors.[1] Consequently, for data across various scales, the number of spatial divisions $L^{(s)}$ differs, and these divisions exhibit a multiplicative relationship across different levels.

$$L^{(s)} = \frac{L}{C^{s-1}}, \quad s = 1, 2, \ldots, S. \tag{1}$$

It is important to note that both S and C are treated as hyperparameters. This design choice is grounded in the observation that real-world multi-scale sensing scenarios often involve a few distinct and classifiable scale levels, such as data from smartphones, vehicles, and drones. Our framework is thus

[1] To simplify the problem, we assume that the inclusion relationship between different scales is identical. For scenarios where these relationships differ, such as $C = 2$ between scale 1 and 2, but $C = 3$ between scale 2 and 3, our model can be adapted by employing different C values across the respective scale transitions.

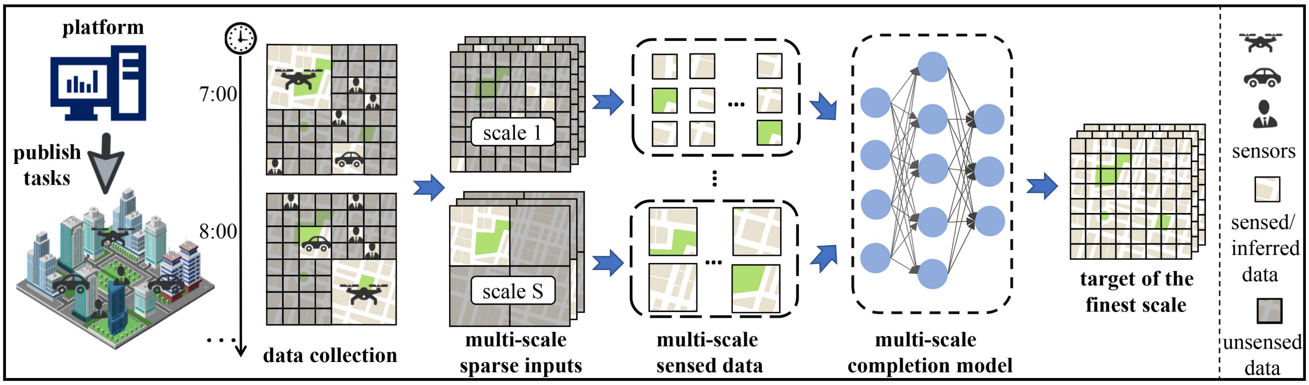


Fig. 3. The workflow of our work: the platform recruits users with different sensing capabilities to collect data for data completion.

designed to be flexible, allowing S and C to be configured to best represent the specific multi-scale structure of a given sensing environment.

Data: For the t^{th} time period, the data captured by users with level s sensing capability in their respective ℓ^{th} subarea is denoted as $x_{t,\ell}^{(s)}$. Unsensed data is recorded as a value of 0.[2] Concurrently, the ground truth is represented as $y_{t,\ell}^{(s)}$. To describe the sensing status at each scale s, we introduce a sensing matrix $\mathbf{M}^{(s)} \in \mathbb{R}^{T \times L^{(s)}}$, where $m_{t,\ell}^{(s)} = 1$ indicates that data at scale s has been sensed during the t^{th} time period and ℓ^{th} subarea; Conversely, $m_{t,\ell}^{(s)} = 0$ indicates it has not been sensed. From this, the sensed data can be expressed as:

$$\mathbf{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(S)}\}, \tag{2}$$

$$\mathbf{X}^{(s)} = \mathbf{Y}^{(s)} \odot \mathbf{M}^{(s)}, \quad s = 1, 2, \ldots, S, \tag{3}$$

where $\mathbf{X}$ represents all the sensed data across different scales, the dot product ($\odot$) represents the element-wise product.

Method: We utilize the completion algorithm $\mathcal{I}()$ to process the sensed multi-scale data $\mathbf{X}$. Since the data at the finest scale is the most difficult to obtain, yet highly demanded in practical applications, our goal is to complete the data at this finest scale ($s = 1$). The completed result is represented as $\hat{\mathbf{Y}}^{(1)} \in \mathbb{R}^{T \times L}$. To quantify the accuracy of this completion, we introduce δ to represent the error between the completed data and the ground truth. This process can be mathematically expressed as:

$$\mathcal{I}(\mathbf{X}) = \hat{\mathbf{Y}}^{(1)} \approx \mathbf{Y}^{(1)}, \tag{4}$$

$$\delta(\hat{\mathbf{Y}}^{(1)}, \mathbf{Y}^{(1)}) = \sum_{i=1}^{T} \sum_{j=1}^{L} \left| y_{i,j}^{(1)} - \hat{y}_{i,j}^{(1)} \right|. \tag{5}$$

B. Problem Formulation

Problem [Data Completion for Multi-scale Sparse Spatio-temporal Data]: Given T time slices, S different scales of data, and L subareas of the finest scale size, we aim to sense data from a limited number of subareas across different scales, utilize this data to reconstruct complete data at the finest scale. In doing so, we strive to minimize the error between the completion results and the ground truth.

$$\min \quad \delta(\hat{\mathbf{Y}}^{(1)}, \mathbf{Y}^{(1)}) = \sum_{i=1}^{T} \sum_{j=1}^{L} \left| y_{i,j}^{(1)} - \hat{y}_{i,j}^{(1)} \right|, \tag{6}$$

$$\text{s.t.} \quad \forall s \in (1, S], \quad \frac{\text{size}(s)}{\text{size}(s-1)} = C. \tag{7}$$

C. Workflow

Our workflow is shown in Fig. 3. We focus on an urban spatio-temporal data sensing task, aiming at acquiring location-specific data over a designated time period in order to infer complete data. Initially, the platform publishes sensing tasks to users equipped with various sensing devices. These users then collect data in specified areas and times, subsequently uploading their data to the platform. After the data collection period concludes, the platform aggregates and categorizes the data according to the sensing scale of each user. Ultimately, this aggregated data is fed into a completion model, which generates the finest-scale completion results.

IV. Method

A. Overall Structure

As shown in Fig. 4, our method is composed of several components: Multi-Scale Embedding Layer, Encoder, Spatial-Temporal Pyramid Construction Module (ST-PC), Decoder, Inference Layer, and Cross-scale Constraint Mechanism.

First, the Multi-Scale Embedding Layer is used to extract initial features from both sensed and unsensed multi-scale input data. The Encoder then processes the sensed data to capture internal correlations. The resulting output is concatenated with the embedded unsensed data to combine the information from both, forming a comprehensive multi-scale representation. To further capture and organize multi-scale information from spatio-temporal data, we specifically design the ST-PC layer to construct a spatio-temporal pyramid structure that aids the decoder in extracting richer multi-scale features. Subsequently, the Decoder works to extract correlations between the unsensed and sensed data, leveraging the Spatio-Temporal

[2] if 0 has a specific meaning, an alternative value will be used.

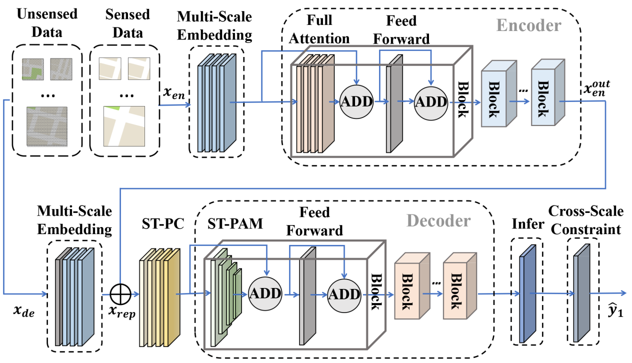


Fig. 4. The structure of our proposed model.

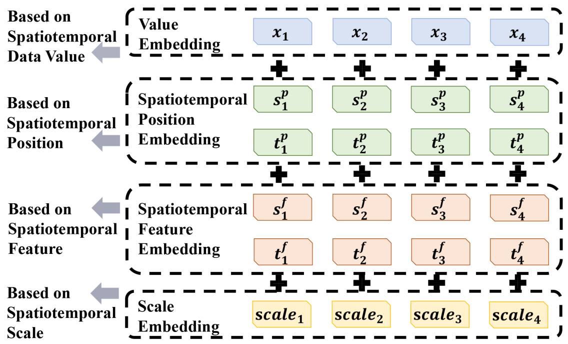

Based on Spatiotemporal Data Value
Value Embedding
x_1 x_2 x_3 x_4
Based on Spatiotemporal Position
Spatiotemporal Position Embedding
s_1^p s_2^p s_3^p s_4^p
t_1^p t_2^p t_3^p t_4^p
Based on Spatiotemporal Feature
Spatiotemporal Feature Embedding
s_1^f s_2^f s_3^f s_4^f
t_1^f t_2^f t_3^f t_4^f
Based on Spatiotemporal Scale
Scale Embedding
$scale_1$ $scale_2$ $scale_3$ $scale_4$

Fig. 5. The embedding layer of our model.

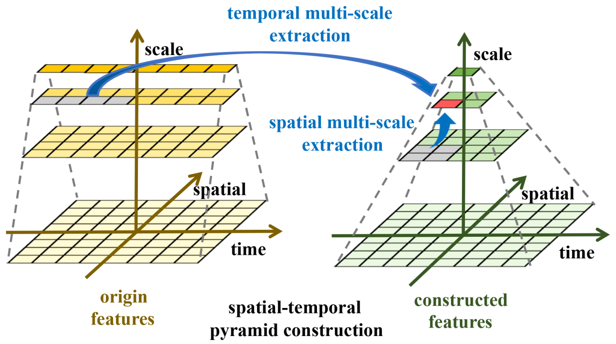

temporal multi-scale extraction
scale
spatial multi-scale extraction
scale
spatial
time
spatial
time
origin features
spatial-temporal pyramid construction
constructed features

Fig. 6. Processes of our spatial-temporal pyramid construction module. The red nodes are constructed using the gray nodes.

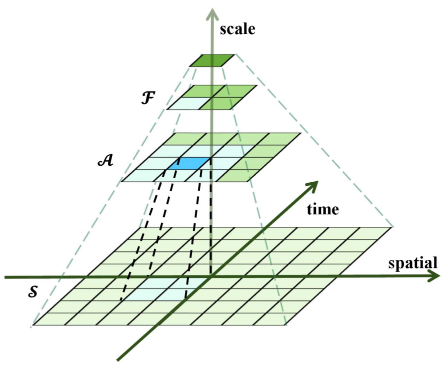


Fig. 7. Structure of the spatial-temporal pyramid attention mechanism. Light blue nodes indicate the nodes that the blue nodes need to focus on.

Pyramid Attention Mechanism (ST-PAM) to enhance this process. Finally, the Cross-Scale Constraint Mechanism is applied to further refine and optimize the completion of the finest-scale data.

B. Multi-Scale Embedding

Data embedding plays a crucial role in completion models, significantly enhancing the model's performance and accuracy. Given that our method processes multi-scale spatio-temporal data inputs, effectively capturing the spatio-temporal characteristics and data scales is key in designing the embedding module. To this end, we first develop a dedicated multi-scale data embedding module, to effectively extract features from multi-scale spatio-temporal data.

As shown in Fig. 5, our model, inspired by the Transformer architecture [27], incorporates value embedding and various forms of positional and contextual embeddings tailored for this task.

1) Value Embedding: For value embedding, we differentiate between sensed and unsensed data. For sensed data, we use one-dimensional convolutions to map the sparse observed values into a d_{model}-dimensional space. For unsensed data, inspired by the Masked Auto Encoder [32], we employ a uniform random learnable vector for its representation. Crucially, to better capture scale-specific imputation patterns, we use a distinct learnable vector for each scale. The final value embedding is represented as $\mathbf{X}_{val}^{(s)} \in \mathbb{R}^{T\times L^{(s)}\times d_{model}}$.

2) Positional Embedding: Given the spatio-temporal nature of our data, we use separate learnable embedding layers for time and space positions. These are represented as $\mathbf{T}_p^{(s)} \in \mathbb{R}^{T\times L^{(s)}\times d_{model}}$ for time-position embeddings and $\mathbf{S}_p^{(s)} \in \mathbb{R}^{T\times L^{(s)}\times d_{model}}$ for space-position embeddings.

3) Contextual Feature Embedding: To further enhance accuracy, we incorporate additional contextual information derived from prior knowledge. For temporal features, including timestamps and holiday indicators, we adopt an approach inspired by Informer [33]. These features, after appropriate normalization, are projected into d_{model}-dimensional vectors using a linear layer. This results in the temporal contextual feature embeddings $\mathbf{T}_f^{(s)}$. For spatial features, our method is similar to that in ST-TransI [23]. Longitude and latitude coordinates are first normalized, then independently projected by separate linear layers, and their resulting embeddings are summed. If Point of Interest (POI) information is available for a location, after one-hot encoding, its d_{model}-dimension embedding is then added to the summed coordinate embeddings to form the final spatial contextual feature embedding $\mathbf{S}_f^{(s)}$.

4) Scale Embedding: Moreover, to accurately perceive the scale of the data, we introduce a scale embedding. This begins by computing the relative scale information as $x_{scale} = C^s/C^S$. Subsequently, this information is mapped to a higher dimensional space using a fully connected layer, resulting in $\mathbf{X}_{scale}^{(s)} \in \mathbb{R}^{T\times L^{(s)}\times d_{model}}$.

The final embedded representation for each spatio-temporal point (t, l) at scale s, denoted as $\mathbf{X}_{emb,t,l}^{(s)}$, is constructed by summing these individual embedding components:

$$\mathbf{X}_{emb}^{(s)} = \mathbf{X}_{val}^{(s)} + \mathbf{S}_{p}^{(s)} + \mathbf{T}_{p}^{(s)} + \mathbf{S}_{f}^{(s)} + \mathbf{T}_{f}^{(s)} + \mathbf{X}_{scale}^{(s)}, \quad (8)$$

Subsequently, based on the sensing matrix $\mathbf{M}^{(s)}$, these pointwise embeddings are partitioned into sets for sensed and unsensed data:

$$\mathbf{X}_{en}^{emb} = \bigcup_{s=1}^{S} \{\mathbf{X}_{emb,t,l}^{(s)} | \mathbf{M}_{t,l}^{(s)} = 1\}, \quad (9)$$

$$\mathbf{X}_{de}^{emb} = \bigcup_{s=1}^{S} \{\mathbf{X}_{emb,t,l}^{(s)} | \mathbf{M}_{t,l}^{(s)} = 0\}, \quad (10)$$

where $\mathbf{X}_{emb,t,l}^{(s)}$ represents the embedding of the spatio-temporal point at time t and subarea l within scale s, and $\mathbf{M}^{(s)}$ is the corresponding sensing matrix.

C. Encoder

After obtaining the multi-scale embeddings, the Encoder processes these embeddings to capture intricate spatio-temporal relationships within the sensed data. It is designed to extract key dependencies across scales while ensuring computational efficiency, setting the stage for subsequent stages of data completion.

Inspired by the Masked AutoEncoder [32], the Encoder focuses exclusively on sparse sensed data, which typically holds the most critical information for spatio-temporal modeling. To capture these essential correlations across scales and time while minimizing computational complexity, the Encoder employs a self-attention mechanism, which dynamically focuses on the most relevant components. Specifically, the self-attention mechanism transforms the multi-scale embeddings of sensed data into query, key, and value matrices:

$$\mathbf{Q} = \mathbf{X}_{en}^{emb}\mathbf{W}^{Q}, \mathbf{K} = \mathbf{X}_{en}^{emb}\mathbf{W}^{K}, \mathbf{V} = \mathbf{X}_{en}^{emb}\mathbf{W}^{V}. \quad (11)$$

Subsequently, the attention scores are calculated using the formula of Scaled Dot-Product Attention:

$$\mathbf{y}_i = \sum_{j=1}^{N} \frac{\exp\left(\mathbf{q}_i \mathbf{k}_j^T / \sqrt{d_k}\right)\mathbf{v}_j}{\sum_{\ell=1}^{N} \exp\left(\mathbf{q}_i \mathbf{k}_\ell^T / \sqrt{d_k}\right)}, \quad (12)$$

where $\mathbf{q}_i$ represents the i-th row of $\mathbf{Q}$, $\mathbf{k}_j^T$ represents the transpose of the j-th row of $\mathbf{K}$, and $\mathbf{v}_j$ represents the j-th row of $\mathbf{V}$, N is the size of the input, and $\sqrt{d_k}$ is the dimension of $\mathbf{K}$, used for scaling the dot product.

This attention mechanism allows the model to focus on the most important components of the sparse sensed data, dynamically adjusting based on learned attention weights. The Encoder processes these features through multiple layers, each refining the spatio-temporal correlations. After passing through all the layers, the final output from the Encoder, denoted as $\mathbf{X}_{en}^{out}$, is prepared for subsequent stages of processing.

$$\begin{aligned}
\mathbf{Z}_{en}^{l,1} &= \text{Norm}\left(\text{FA}(\mathbf{X}_{en}^{l-1}) + \mathbf{X}_{en}^{l-1}\right), \\
\mathbf{Z}_{en}^{l,2} &= \text{Norm}\left(\text{FFN}(\mathbf{Z}_{en}^{l,1}) + \mathbf{Z}_{en}^{l,1}\right), \\
\mathbf{X}_{en}^{l} &= \mathbf{Z}_{en}^{l,2}, l = 1, \ldots, N, \\
\mathbf{X}_{en}^{0} &= \mathbf{X}_{en}^{emb}, \mathbf{X}_{en}^{out} = \mathbf{X}_{en}^{N}, \quad (13)
\end{aligned}$$

where FA() represents the self-attention mechanism, FFN() represents the feedforward network, and Norm() represents the LayerNorm. $\mathbf{X}_{en}^{out}$ represents the final encoded output after processing through all layers. The Encoder efficiently captures key relationships in sparse multi-scale spatio-temporal data, setting the stage for the next phase of data reconstruction.

Before transitioning to the next module, it is crucial to address that the Encoder focuses solely on sensed data, which lacks the context provided by the unsensed data. We create a comprehensive multi-scale spatio-temporal representation by combining the Encoder's output, $\mathbf{X}_{en}^{out}$, with the embedded unsensed data. This process can be expressed as:

$$\mathbf{X}_{rep} = Resort(Concat(\mathbf{X}_{en}^{out}, \mathbf{X}_{de}^{emb})), \quad (14)$$

where $Concat(\mathbf{X}_{en}^{out}, \mathbf{X}_{de}^{emb})$ represents concatenation $\mathbf{X}_{en}^{out}$ and $\mathbf{X}_{de}^{emb}$ along the node dimension. Subsequently, $Resort(.)$ places the combined embeddings back into their original locations. This comprehensive representation lays the foundation for further exploration of spatio-temporal information.

D. Spatial-Temporal Pyramid Construction Module (ST-PC)

After obtaining the output from the Encoder, we need to construct a structure that effectively captures the multi-scale spatio-temporal relationships. Spatio-temporal data is complex, often displaying patterns across various scales and different temporal contexts. Basic approaches to handling multi-scale data often struggle with balancing fine-scale and coarse-scale information, leading to a loss of critical details or inefficient computation. To address these challenges, a robust structure is required—one that effectively integrates information across scales while preserving the rich characteristics.

For this purpose, we have selected a pyramid structure. The pyramid structure is particularly advantageous because it allows for the simultaneous consideration of fine-grained and coarse-grained patterns. In the spatial dimension, this enables the model to capture both local details and broader patterns, such as how data collected by handheld devices in small areas relates to broader trends captured by drones covering larger regions. In the temporal dimension, it models short-term fluctuations and long-term trends, ensuring that the model can relate immediate events to broader temporal behaviors, like daily variations compared to weekly or monthly patterns. By integrating these spatial and temporal scales, the pyramid structure effectively combines both trends, enabling comprehensive spatio-temporal data reconstruction.

Building on these advantages, we introduce the Spatio-Temporal Pyramid Construction Module (ST-PC) to effectively implement this structure. The ST-PC module is specifically designed to extract and integrate features across multiple spatio-temporal scales, effectively addressing the challenges posed by sparse input data. This module constructs a pyramid structure where each layer represents a specific scale, with coarser scales at higher levels. In each layer, the rows correspond to time and the columns correspond to space, with a direct correspondence between layers: each coarser-scale node encapsulates the combined information of $C \times C$ finer-scale nodes in the layer below, summarizing detailed spatio-temporal patterns.

To effectively implement this pyramid structure, the ST-PC module is divided into two main parts: Temporal Multi-Scale Extraction and Spatial Multi-Scale Extraction. As shown in Fig. 6, we first focus on constructing a pyramid structure along the temporal dimension. Starting from the finest temporal resolution, we aggregate the information across time while maintaining the spatial distribution. To achieve this, we employ a one-dimensional convolutional layer with a kernel size of C^{s-1} and a stride of C^{s-1} to construct nodes at the corresponding scale:

$$x_{t-pc_{t,\ell}} = \sum_{i=1}^{C^{s-1}} w_i x_{rep}{}_{(t-1)C+i,\ell}^{(s)} + b_i, \tag{15}$$

where w_i represents the i-th element of the one-dimensional convolution kernel $\mathbf{w} \in \mathbb{R}^C$, and b_i represents the i-th element of the bias $\mathbf{b} \in \mathbb{R}^C$.

Since our spatio-temporal data is highly sparse, constructing effective representations at different scales is crucial to ensure that no critical information is lost during the process. To achieve this, we aggregate these finer spatio-temporal nodes to form coarser representations, ensuring that we capture as much relevant information as possible. To maximize data utilization, we design the Spatial Multi-Scale Extraction part to enhance data utilization. This part uses a two-dimensional convolutional layer with a kernel size and stride of C^{s-1} to utilize smaller scale nodes to construct coarser nodes:

$$x_{s-pc_{t,\ell}} = \sum_{i=1}^{C^{s-1}} \sum_{j=1}^{C^{s-1}} w_{i,j} x_{t-pc}{}_{(t-1)C+i,(\ell-1)C+j}^{(s-1)} + b_{i,j}, \tag{16}$$

where $w_{i,j}$ represents the element of row i and column j in the two-dimensional convolution kernel $\mathbf{W} \in \mathbb{R}^{C^{s-1} \times C^{s-1}}$, and b_i represents the row i and column j in the bias $\mathbf{B} \in \mathbb{R}^{C^{s-1} \times C^{s-1}}$.

Ultimately, we concatenate the results from these two parts and map their dimensions to d_{model} through a fully connected layer, thus forming the newly constructed node set at scale s,

$$\mathbf{X}_{pc}^{(s)} = Linear\left(Concat(\mathbf{X}_{t-pc}^{(s)}, \mathbf{X}_{s-pc}^{(s)})\right). \tag{17}$$

In this manner, we construct the entire spatio-temporal pyramid structure:

$$\mathbf{X}_{pc} = \{\mathbf{X}_{pc}^{(1)}, \mathbf{X}_{pc}^{(2)}, \ldots, \mathbf{X}_{pc}^{(S)}\}, \tag{18}$$

this structure serves as the input for the decoder.

E. Decoder

After constructing the pyramid structure, the next step is to facilitate direct interactions between the different scales within the Decoder. Similar to the Encoder, the Decoder processes the input through multiple layers, maintaining the hierarchical structure. However, unlike the Encoder, the Decoder incorporates the Spatio-Temporal Pyramid Attention Mechanism (ST-PAM), which is specifically designed to capture multi-scale dependencies by enabling direct interactions across different scales. The Decoder's operations can be formulated as:

$$\begin{aligned} c\mathbf{X}_{de}^{l,1} &= \text{Norm}\left(\text{STP}(\mathbf{X}_{de}^{l-1}) + \mathbf{X}_{de}^{l-1}\right), \\ \mathbf{X}_{de}^{l,2} &= \text{Norm}\left(\text{FFN}(\mathbf{X}_{de}^{l,1}) + \mathbf{X}_{de}^{l,1}\right), \\ \mathbf{X}_{de}^{l} &= \mathbf{X}_{de}^{l,2}, l = 1, \ldots, N, \\ \mathbf{X}_{de}^{0} &= \mathbf{X}_{pc}, \mathbf{X}_{de}^{out} = \mathbf{X}_{de}^{N}, \end{aligned} \tag{19}$$

where STP() represents the ST-PAM, and $\mathbf{X}_{de}^{out}$ represents the final decoded output. Finally, the trained features are input into the inference layer to obtain the finest-scale completion results. Next, we will provide a detailed explanation of the ST-PAM, the core component enabling multi-scale interactions within the Decoder.

F. Spatial-Temporal Pyramid Attention Mechanism (ST-PAM)

Within the Decoder, the ST-PAM plays a central role in directly modeling the interactions across different scales. By leveraging the pyramid structure established in the previous module, ST-PAM allows the model to effectively capture both fine-scale and coarse-scale dependencies in spatio-temporal data. This mechanism is integral to the Decoder's ability to process the multi-scale information gathered from the Encoder and construct accurate spatio-temporal representations.

As shown in Fig. 7, in the ST-PAM, we emphasize the importance of capturing two types of interactions: intra-scale connections and inter-scale connections. Intra-scale connections focus on capturing localized patterns within the same scale, allowing the model to learn detailed spatio-temporal correlations. On the other hand, inter-scale connections bridge the relationships between finer and coarser scales, ensuring that detailed information flows smoothly across the pyramid's hierarchy and larger-scale trends are maintained. By considering both types of connections, ST-PAM effectively captures the complex multi-scale spatial and temporal relationships.

ST-PAM handles intra-scale connections by attending to neighboring nodes within the same layer. Specifically, for each node at a given scale, the mechanism considers the closest A^2 nodes, where A is a hyperparameter, representing the number of adjacent spatio-temporal steps considered along both the temporal and spatial dimensions. Due to the inherent spatio-temporal dependencies [34], nodes that are closer in time and space tend to have stronger correlations. By selecting A^2 neighboring nodes, the model captures localized patterns more effectively, leveraging both temporal and spatial proximity to enhance the understanding of fine-grained details.

Simultaneously, to establish effective inter-scale connections, we focus on the relationship between coarse-scale and finer-scale nodes. Specifically, each coarse-scale node is connected to the C^2 finer-scale nodes that form its composition, enabling the model to aggregate detailed information from finer levels into the coarser representation. Additionally, connections are established between coarser-scale nodes and their counterparts in adjacent layers, ensuring the model retains both the intricate details from finer scales and the broader context across space and time, harmonizing detailed insights with overarching trends. For regular sensing scenarios, the ST-PAM can be formulated as follows:

$$\mathcal{A}_{t,\ell}^{(s)} = \left\{ \mathbf{x}_{i,j}^{(s)} : \begin{array}{l} t - \frac{A-1}{2} \leq i \leq t + \frac{A-1}{2}, \\ l - \frac{A-1}{2} \leq j \leq \ell + \frac{A-1}{2}, \end{array} \right\},$$

$$\mathcal{S}_{t,\ell}^{(s)} = \left\{ \mathbf{x}_{i,j}^{(s-1)} : \begin{array}{l} (t-1)C < i \leq tC, \\ (\ell-1)C < j \leq \ell C, \end{array} \right\},$$

$$\mathcal{F}_{t,\ell}^{(s)} = \left\{ \mathbf{x}_{i,j}^{(s+1)} : i = \left\lceil \frac{t}{C} \right\rceil, j = \left\lceil \frac{\ell}{C} \right\rceil \right\},$$

$$\mathcal{M}_{t,\ell}^{(s)} = \mathcal{A}_{t,\ell}^{(s)} \cup \mathcal{F}_{t,\ell}^{(s)} \cup \mathcal{S}_{t,\ell}^{(s)}, \tag{20}$$

where $\mathcal{A}_{t,\ell}^{(s)}$ represents the attention nodes within the same scale, $\mathcal{S}_{t,\ell}^{(s)}$ represents the finer-scale nodes that form the current node, and $\mathcal{F}_{t,\ell}^{(s)}$ represents the coarser-scale nodes associated with the current node.

For irregular sensing scenarios (e.g., traffic management), we redefine only the intra-scale neighborhood $\mathcal{A}$ to align the spatial neighborhood with the actual topology:

$$\mathcal{A}_{t,\ell}^{(s)} = \left\{ \mathbf{x}_{i,u}^{(s)} : \begin{array}{c} t - \frac{A-1}{2} \leq i \leq t + \frac{A-1}{2}, \\ u \in \mathcal{N}_A(v), \end{array} \right\}, \tag{21}$$

where the spatial neighbor set $\mathcal{N}_A(v)$ is defined by

$$\mathcal{N}_A(v) = \underset{S \subseteq \mathcal{V}, |S| = A}{\operatorname{argmin}} \sum_{p \in S} d_{\text{graph}}(p, v), \tag{22}$$

and $d_{\text{graph}}(\cdot, \cdot)$ denotes the shortest path distance between two sensors. The full attention mechanism is defined as:

$$\mathbf{y}_{t,\ell}^{(s)} = \sum_{m \in \mathcal{M}_{t,\ell}^{(s)}} \frac{\exp\left(\mathbf{q}_{t,\ell} \mathbf{k}_m^T / \sqrt{d_k}\right) \mathbf{v}_m}{\sum_{m \in \mathcal{M}_{t,\ell}^{(s)}} \exp\left(\mathbf{q}_{t,\ell} \mathbf{k}_m^T / \sqrt{d_k}\right)}. \tag{23}$$

ST-PAM significantly reduces computational complexity while maintaining high performance, with a computational complexity of $O(LT)$ and a maximum signal traversal path length of $O(1)$ under specific conditions.

G. Cross-Scale Constraint

In order to further refine the data completion process, especially at the finest scale, it is crucial to incorporate the inherent mathematical relationships between data at different scales. Operations such as summation and averaging provide valuable insights from coarser scale data that can be leveraged to correct and optimize the completion at finer scales. To address this, we propose a novel Cross-Scale Constraint method aimed at guiding the completion process across multiple scales, ensuring consistency and accuracy.

Specifically, we introduce a loss function $\mathcal{L}$ as follows:

$$\mathcal{L} = \alpha \mathcal{L}_{acc}\left(\widehat{\mathbf{Y}}^{(1)}, \mathbf{Y}^{(1)}\right) + (1-\alpha)\mathcal{L}_{rel}\left(\widehat{\mathbf{Y}}^{(1)}, \mathbf{Y}\right), \tag{24}$$

where α is a weight parameter between 0 and 1. $\mathcal{L}_{acc}$ represents the discrepancy between the data completion results and the ground truth on the finest scale, quantified in this study using the Mean Squared Error (MSE) function. $\mathcal{L}_{rel}$ represents the discrepancy between the data perceived at coarser scales and the completed data, calculated as follows:

Algorithm 1: Multi-Scale Data Completion.

```
Input: $A, C, T, L, N, S, \mathbf{X}$
Output: $\hat{\mathbf{Y}}^{(1)}$
$count \leftarrow 0$;
while not convergent and $count <$ MAX_ITER do
    Embedding:
    Divide $\mathbf{X}$ into sensed part $\mathbf{X}_{en}$ and unsensed part $\mathbf{X}_{de}$;
    Get $\mathbf{X}_{val}^{(s)}, \mathbf{S}_p^{(s)}, \mathbf{T}_p^{(s)}, \mathbf{S}_f^{(s)}, \mathbf{T}_f^{(s)}, \mathbf{X}_{scale}^{(s)}$ for both $\mathbf{X}_{en}$ and $\mathbf{X}_{de}$;
    Calculate $\mathbf{X}_{en}^{emb}, \mathbf{X}_{de}^{emb}$ by (9) and (10);
    Encoder:
    $\mathbf{X}_{en}^0 \leftarrow \mathbf{X}_{en}^{emb}$;
    for $l = 1$ to $N$ $encoders$ do
        Calculate $\mathbf{X}_{en}^l$ by (13);
     $\mathbf{X}_{en}^{out} \leftarrow \mathbf{X}_{en}^N$;
     Decoder:
     Calculate $\mathbf{X}_{rep}$ by (14);
     Calculate $\mathbf{X}_{t-pc}^{(s)}$ and $\mathbf{X}_{s-pc}^{(s)}$ by (15) and (16);
     Calculate $\mathbf{X}_{pc}$ by (17) and (18);
     $\mathbf{X}_{de}^0 \leftarrow \mathbf{X}_{pc}$;
     for $l = 1$ to $N$ $decoders$ do
        Calculate $\mathbf{X}_{de}^l$ by (19);
     $\mathbf{X}_{de}^{out} \leftarrow \mathbf{X}_{de}^N$;
     Get $\hat{\mathbf{Y}}^{(1)}$ by inputting $\mathbf{X}_{de}^{out}$ to infer network.
     Calculate and reduce $\mathcal{L}$ by (24);
     $count \leftarrow count + 1$
 return $\hat{\mathbf{Y}}^{(1)}$
```

$$\mathcal{L}_{rel} = \sum_{s=2}^{S} \mathcal{L}_{mse}(Agg\_s(\widehat{\mathbf{Y}}^{(1)}) \odot \mathbf{M}^{(s)}, \mathbf{Y}^{(s)} \odot \mathbf{M}^{(s)}), \tag{25}$$

where $\odot$ denotes element-wise multiplication, $\mathbf{M}^{(s)}$ represents the sampling matrix at scale s, and $Agg\_s(.)$ transforms the finest-scale result to scale s.[3]

Through this approach, our method not only considers the accuracy at the finest scale but also leverages coarser scale data to enhance the overall quality and precision of the completion. We show our whole work flow in Algorithm 1.

V. Theoretical Analysis

Definition 1. Maximum Path Length: The maximum path length is the maximum number of sequential processing steps required for information to propagate from any input position to any other input position within the sequence.

[3] If the task is concerned with the quality of the output at multiple scales, we can treat it as a multitasking optimization loss, and the weighting factors for each scale's loss can be adjusted manually for the more concerned scales.

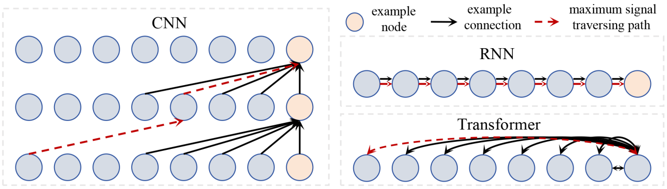


Fig. 8. Maximum path length of commonly used models.

To provide a clearer intuition for this concept, Fig. 8 visually contrasts the maximum path lengths required by different common architectures to relate distant signals in a sequence. A standard Recurrent Neural Network (RNN) processes information sequentially, resulting in a maximum path length of $O(N)$ for a sequence of length N. A typical Convolutional Neural Network (CNN) expands its receptive field by stacking layers, leading to a path length that is proportional to its depth, often on the order of $O(\log N)$ with dilated convolutions. In contrast, a standard Transformer can achieve an $O(1)$ maximum path length, as its self-attention mechanism allows any input position to directly interact with any other within a single layer.

Theorem 1: When the given A, C, T, L, N, S satisfy the (26), the coarsest scale node can obtain the global receptive field after stacking N layers of ST-PAM layers.

$$\frac{\max(T, L)}{C^{S-1}} - 1 \leq \frac{(A-1)N}{2}. \tag{26}$$

Proof: Let S denote the number of scales, C denote the number of nodes of the coarser time scale s containing the finer scale s-1, and the same at the spatial scale. Clearly, the coarsest scale node is composed of $C^{2(S-1)}$ finest scale nodes. Without sacrificing generalizability, we assume that both L and T are divisible by C^{S-1}. Then the number of coarsest scale nodes is $\frac{T \cdot L}{C^{2(S-1)}}$. Since for each attention layer, at the same scale, each node can be connected as far away from $\frac{(A-1)}{2}$, the distance between the leftmost and rightmost node on the coarsest scale is $\frac{2\max(T,L)}{(A-1)C^{S-1}}$. Therefore, when the number of stacked attention layers satisfies (27), the coarsest-scale node is within the receptive field of all nodes at the current scale.

$$N \geq \frac{2\max(T, L)}{(A-1)C^{S-1}}. \tag{27}$$

Moreover, due to ST-PC, the coarsest scale nodes can be seen as summaries of the corresponding finer scale nodes. As a result, when (26) is satisfied, all nodes at the coarsest scale have a global receptive field. □

Theorem 2: The time and space complexity for the Spatial-Temporal Pyramid Attention mechanism is $O(LT)$.

Proof: Let $L^{(s)}$ and $T^{(s)}$ denote the number of spaces and the number of time slices of a node at scale s. Then,

$$L^{(s)} = \frac{L}{C^{s-1}}, T^{(s)} = \frac{T}{C^{s-1}}, 1 \leq s \leq S. \tag{28}$$

For the node $n_{t,\ell}^{(s)}$ in a pyramid graph, the number of dot products it has as queries $P_{t,\ell}^{(s)}$ can be decomposed into two parts:

$$P_{t,\ell}^{(s)} = P_{t,\ell\ in}^{(s)} + P_{t,\ell\ off}^{(s)}, \tag{29}$$

where $P_{t,\ell\ in}^{(s)}$ and $P_{t,\ell\ off}^{(s)}$ denote the in-scale and off-scale part respectively. According to the structure of our attention mechanism, we can obviously conclude the following:

$$P_{t,\ell\ in}^{(s)} \leq A * A, P_{t,\ell\ off}^{(s)} \leq C * C + 1. \tag{30}$$

Therefore, the total number of dot products at scale s is:

$$\begin{aligned} P^{(s)} &= \sum_{\ell=1}^{L^{(s)}} \sum_{t=1}^{T^{(s)}} \left(P_{t,\ell\ in}^{(s)} + P_{t,\ell\ off}^{(s)} \right) \\ &\leq L^{(s)} T^{(s)} (A * A + C * C + 1). \end{aligned} \tag{31}$$

In summary, the total number of dot products to be computed in the entire attention mechanism is:

$$\begin{aligned} P &= \sum_{s=1}^{S} P^{(s)} \\ &\leq LT(A^2+1) + \cdots + L^{(S)}T^{(S)}(A^2 + C^2 + 1) \\ &< LT((A^2+2)\sum_{s=1}^{S} C^{-2(s-1)} + 1). \end{aligned} \tag{32}$$

Thus, the complexity of the proposed attention is:

$$\begin{aligned} O(P) &\leq O(LT \left((A * A + 2) \sum_{s=1}^{S} C^{-(s-1)} + 1 \right) \\ &= O\left(LT(A * A + 2) \sum_{s=1}^{S} C^{-(s-1)} \right) \\ &= O\left(\frac{(A+2)LT(1 - C^{-S})}{1 - C^{-1}} \right) \\ &= O\left((A+2)LT\right) = O(ALT) = O(LT). \end{aligned} \tag{33}$$

As a result, the complexity is $O(LT)$. □

Theorem 3: When the given hyperparameters satisfy (34), the distance between any two nodes in our proposed model network is $O(1)$.

$$\sqrt[S-1]{\max(L, \mathrm{T})} \geq C \geq \sqrt[S-1]{\frac{\max(L, \mathrm{T})}{\sqrt{\frac{(A-1)N}{2} + 1}}}. \tag{34}$$

Proof:

$$\begin{aligned} O(L_{max}) &= O\left(2(S-1) + \frac{2\left(\max\left(T^{(S)}, L^{(S)}\right) - 1\right)}{A-1} \right) \\ &= O\left(2(S-1) + \frac{2\left(\max\left(\frac{T}{C^{S-1}}, \frac{L}{C^{S-1}}\right) - 1\right)}{A-1} \right) \\ &= O(2(S-1) + N) \\ &= O(S+N). \end{aligned} \tag{35}$$

TABLE I
COMPARISON OF COMPUTATIONAL OVERHEAD AND EFFICIENCY

Method	Computational Complexity	Maximum Path Length
CNN	$O(LT)$	$O(\log(LT))$
RNN	$O(LT)$	$O(LT)$
Transformer	$O((LT)^2)$	$O(1)$
Our	$O(LT)$	$O(1)$

Since A, S, N are given constants, the maximum signal traversal path length is $O(1)$. □

Table I highlights the theoretical advantages of our approach. While a standard Transformer achieves an optimal $O(1)$ maximum path length, its quadratic $O((LT)^2)$ complexity is a significant drawback for long sequences. Conversely, CNNs and RNNs have more efficient linear complexity but suffer from longer path lengths, hindering long-range dependency modeling. Our method uniquely achieves both an optimal $O(1)$ path length and an efficient linear computational complexity of $O(LT)$, combining the primary strengths of these common architectures.

VI. EXPERIMENTS

A. Setting

1) Datasets: To better validate the performance of our model, we conduct experiments on five datasets from three real-world scenarios: Air-Quality, Weather and Traffic.

Air-Quality[4] data set contains data such as PM2.5, NO2, and O3 from 2017 to 2018, recorded hourly at 35 air monitoring stations in Beijing, China. We select $\mathbf{PM2.5}$, $\mathbf{NO2}$, and $\mathbf{O3}$ from the first 32 stations as the experimental dataset.

Weather[4] data set contains meteorological data, such as temperature, humidity and wind speed of different geographic grids from 2017 to 2018 from more than 800 air monitoring stations in London, with each grid recording data every hour. We select *Humidity* and *Wind Speed* as the experimental datasets, retain the original data in an 8 x 4 grid structure.

Traffic[5] data set contains the capacity, speed, flow and other data of more than 400 detectors on the California Highway in the USA, with each detector recording data every 5 minutes. We select *PEMS03* as the experimental datasets, and select the first 32 stations as experimental data.

TaxiBJ [35] data contains real-world crowd flow records collected from taxicab GPS monitors in Beijing, China. It divides the region into a 32×32 grid, with each cell reporting flow information at 30-minute intervals across four distinct time periods (P1–P4). For our experiments, we select **P1**, **P2**, **P3** and **P4** as the experimental datasets.

2) Baselines: We categorize the existing methods into four categories: Single-Scale Spatial Completion, Single-Scale Temporal Interpolation, Single-Scale Spatio-Temporal Completion and Multi-Scale Spatio-Temporal completion methods. Single-scale methods operate at a uniform resolution, while multi-scale methods handle varying input scales.

Single-Scale Spatial Completion Methods:

- *MF [36]:* A classical matrix factorization method to handle sparse data by leveraging low-rank approximations.
- *DMF [12]:* A method combining matrix factorization with deep learning for spatial data completion.
- *GCN [37]:* Utilizes graph convolutional networks to capture spatial dependencies in irregular data structures.
- *FFDNet [38]:* A CNN-based method utilizing denoising techniques for spatial data completion.

Single-Scale Temporal Interpolation Methods:

- *Mamba [39]:* A linear-time sequence foundation model, offering efficient temporal data processing.
- *SAITS [40]:* A multivariate time-series imputation method based on diagonally-masked self-attention.
- *Informer [33]:* A sparse attention model that captures long-range temporal dependencies efficiently.
- *TimesNet [41]:* A general time series model that extracts multi-scale temporal patterns effectively.

Single-Scale Spatio-Temporal Completion Methods:

- *CGAN [42]:* Uses generative adversarial networks to complete data based on conditional inputs.
- *WaveNet [43]:* A Graph-based model for spatial-temporal forecasting via adaptive dependency learning.
- *ST-BGMC [44]:* A low-rank matrix completion method incorporating spatio-temporal constraints.
- *ST-TransI [23]:* A Transformer-based model capturing spatio-temporal correlations for improved completion.
- *PDFormer [45]:* A method for spatio-temporal modeling via propagation-delay-aware dynamic self-attention.
- *ImputeFormer [46]:* A low-rankness-induced Transformer model for spatio-temporal data imputation balancing inductive bias and expressivity.

Multi-Scale Spatio-Temporal Completion Methods:

- *FULL-Attn:* A variant of our proposed method that replaces the ST-PC and ST-PAM modules with self-attention mechanisms for handling multi-scale data.

To address the limitation of single-scale models in handling multi-scale data, we ensure fairness by adjusting the data input. Specifically, the single-scale models are provided with input data equivalent to the total amount of data collected across all scales in the multi-scale models.[6]

3) Data Preprocessing: We preprocess the data to emulate real-world multi-scale data collection scenarios. First, for regular sensing scenarios (e.g., Weather), sensors are already arranged on a grid, so we aggregate neighboring grid cells to create coarser scales. For irregular sensing scenarios (e.g., Traffic), we first project each sensor onto a two-dimensional geographic map and cluster spatially contiguous nodes with strong topological connections into elementary regions. The subsequent aggregation of these regions into coarser scales is defined through a manual but principled process, primarily guided by geographic proximity and the desired hierarchical

[4] https://www.kdd.org/kdd2018/kdd-cup

[5] http://pems.dot.ca.gov.

[6] Note: While we ensure fairness in our evaluation by compensating single-scale models with the total data input across all scales, their actual performance in real-world applications would likely be worse, as they cannot inherently handle multi-scale data effectively.

TABLE II
COMPLETION PERFORMANCE WITH EQUAL SENSED DATA QUANTITY ACROSS MULTIPLE DATASETS

Data	Air-Quality						Weather				Traffic		TaxiBJ							
Set	PM2.5		NO2		O3		Humidity		Wind Speed		PEMS03		P1		P2		P3		P4	
Models	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
MF	0.406	0.366	0.589	0.536	0.457	0.448	0.052	0.151	0.332	0.379	0.208	0.300	0.704	0.540	0.642	0.522	0.560	0.470	0.565	0.458
DMF	0.921	0.603	0.904	0.715	0.873	0.667	0.370	0.452	0.941	0.665	0.976	0.787	1.029	0.728	0.835	0.662	0.835	0.667	0.805	0.638
GCN	0.599	0.492	0.598	0.572	0.583	0.567	0.556	0.570	0.511	0.475	0.536	0.510	0.595	0.553	0.502	0.503	0.500	0.489	0.496	0.475
FFDNet	0.411	0.432	0.511	0.562	0.388	0.481	0.040	0.147	0.299	0.390	0.227	0.360	0.517	0.487	0.437	0.440	0.437	0.426	0.420	0.423
Mamba	0.291	0.312	0.371	0.451	0.285	0.372	0.040	0.150	0.156	0.282	0.098	0.213	0.395	0.369	0.325	0.339	0.332	0.333	0.320	0.331
SAITS	0.329	0.332	0.429	0.481	0.331	0.377	0.032	0.130	0.156	0.283	0.089	0.203	0.381	0.358	0.313	0.322	0.312	0.309	0.303	0.310
Informer	0.302	0.327	0.370	0.449	0.299	0.377	0.038	0.150	0.179	0.311	0.098	0.223	0.397	0.377	0.335	0.353	0.337	0.339	0.318	0.329
TimesNet	0.267	0.315	0.356	0.447	0.268	0.366	0.030	0.130	0.135	0.262	0.081	0.190	0.403	0.374	0.326	0.336	0.333	0.329	0.328	0.329
CGAN	0.599	0.492	0.598	0.572	0.583	0.567	0.556	0.570	0.511	0.475	0.536	0.510	0.700	0.570	0.588	0.518	0.646	0.541	0.622	0.544
WaveNet	0.678	0.546	0.675	0.644	0.429	0.443	0.028	0.121	0.157	0.279	0.079	0.189	0.442	0.405	0.373	0.378	0.369	0.365	0.358	0.350
ST-BGMC	0.387	0.365	0.488	0.516	0.473	0.477	0.083	0.203	0.335	0.402	0.189	0.295	0.544	0.490	0.493	0.484	0.458	0.439	0.452	0.425
ST-TransI	0.344	0.346	0.397	0.444	0.467	0.510	0.016	0.082	0.112	0.223	0.071	0.171	0.395	0.426	0.335	0.317	0.332	0.338	0.347	0.340
PDFormer	0.248	0.296	0.451	0.489	0.315	0.385	0.016	0.085	0.113	0.203	0.084	0.200	0.374	0.348	0.304	0.313	0.304	0.305	0.298	0.303
ImputeFormer	0.339	0.323	0.458	0.484	0.373	0.405	0.020	0.098	0.131	0.244	0.076	0.179	0.375	0.349	0.311	0.313	0.307	0.294	0.298	0.291
FULL-Attn	0.395	0.363	0.281	0.349	0.286	0.333	0.019	0.081	0.092	0.202	0.088	0.190	0.629	0.523	0.555	0.481	0.537	0.462	0.489	0.438
Our	**0.184**	**0.217**	**0.265**	**0.336**	**0.222**	**0.287**	**0.015**	**0.075**	**0.084**	**0.188**	**0.055**	**0.147**	**0.348**	**0.334**	**0.271**	**0.288**	**0.275**	**0.275**	**0.273**	**0.277**

structure. For example, with $C = 2$ and $S = 3$, we downsample the original data by factors of 2 and 4 to simulate three distinct scales. The downsampling process is tailored to the nature of the data: for the Air-Quality and Weather datasets, we apply averaging, while for the Traffic dataset, we use summation to preserve the appropriate scale relationships.

4) Experimental Settings: In our experiment, the datasets are divided into training, validation, and test sets in a $7 : 2 : 1$ ratio. We normalize the data before feeding it into the model. For model training, we use ADAM as the optimizer with an initial learning rate of 0.001. The batch size is set to 16, and training proceeds for 150 epochs. All experiments are conducted using a single NVIDIA GeForce RTX 3090 GPU. We use Mean Squared Error (MSE) and Mean Absolute Error (MAE) as our evaluation metrics.

Regarding the hyperparameter settings, in our model d_{model} is set to 64, d_k to 256, C, A, S to 2, 3, 4, and T to 16, respectively. The number of encoders and decoders are both set to 3. These parameters comply with the (34).

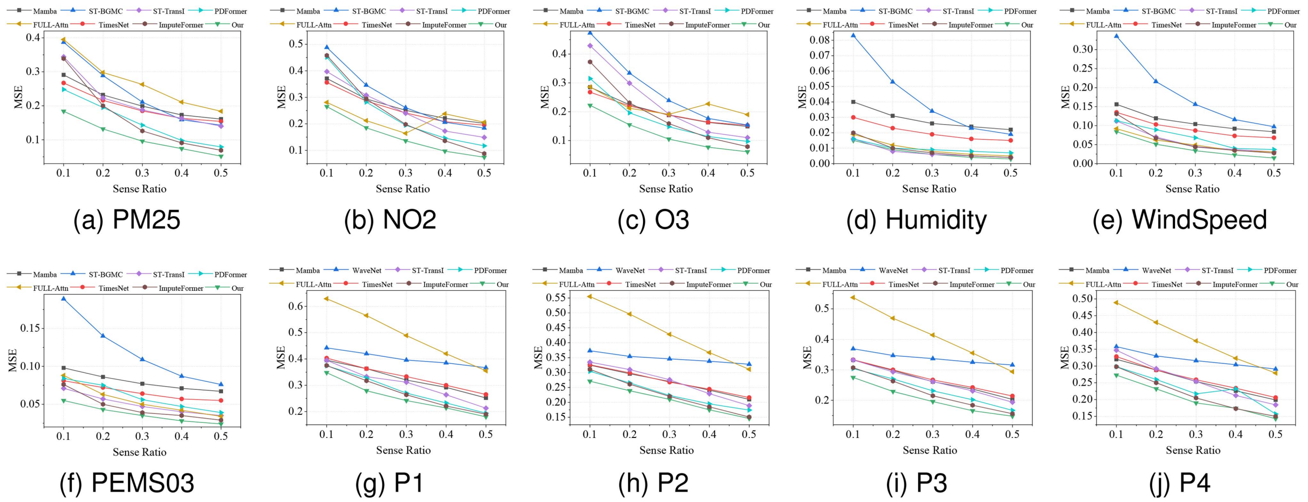
B. Completion Performance

1) Performance With Equal Sensed Data Quantity: We conduct experiments on ten real-world datasets, ensuring equal sensed data across all models. For multi-scale models, the sense ratio for each scale is set to 0.1. Here, sense ratio refers to the proportion of actual collected spatio-temporal points to the total number of spatio-temporal points. Single-scale models receive input data equivalent to the total sensed data from all scales in the multi-scale models for a fair comparison.

As shown in Table II, our method consistently outperforms the baseline algorithms across ten datasets. Spatial data completion methods generally perform well on datasets with relatively stable spatial patterns, such as Weather. However, they struggle with more complex, high-dimensional datasets like Air-Quality, which present more variable temporal dependencies. Temporal interpolation excels at capturing long-term dependencies, but they face challenges in handling intricate spatio-temporal interactions, particularly in more complex datasets like Traffic and TaxiBJ. Spatio-temporal completion methods show better performance in correlating spatial and temporal data but still struggle with multi-scale relationships. Notably, models like CGAN and DMF face convergence issues and fail to adapt to sparse, high-dimensional data, resulting in suboptimal outcomes. WaveNet, while effective on traffic datasets, shows comparatively less robust performance on environmental datasets, potentially due to its convolutional architecture being more adaptable to structured local dependencies. In contrast, our method consistently outperforms these baselines by effectively capturing the relationships across different scales. Although the FULL-Attn method attempts to capture multi-scale relationships, it suffers from redundancy when capturing cross-scale relationships, leading to inferior results compared to our approach.

2) Performance Under Varying Finest-Scale Sense Ratio: We conduct experiments to evaluate how varying the sense ratio at the finest scale affects model performance. In this part, the finest-scale sense ratio ranges from 0.1 to 0.5, while keeping other scales at 0.1. To ensure fairness, single-scale models receive the same total sensed data as the multi-scale models.

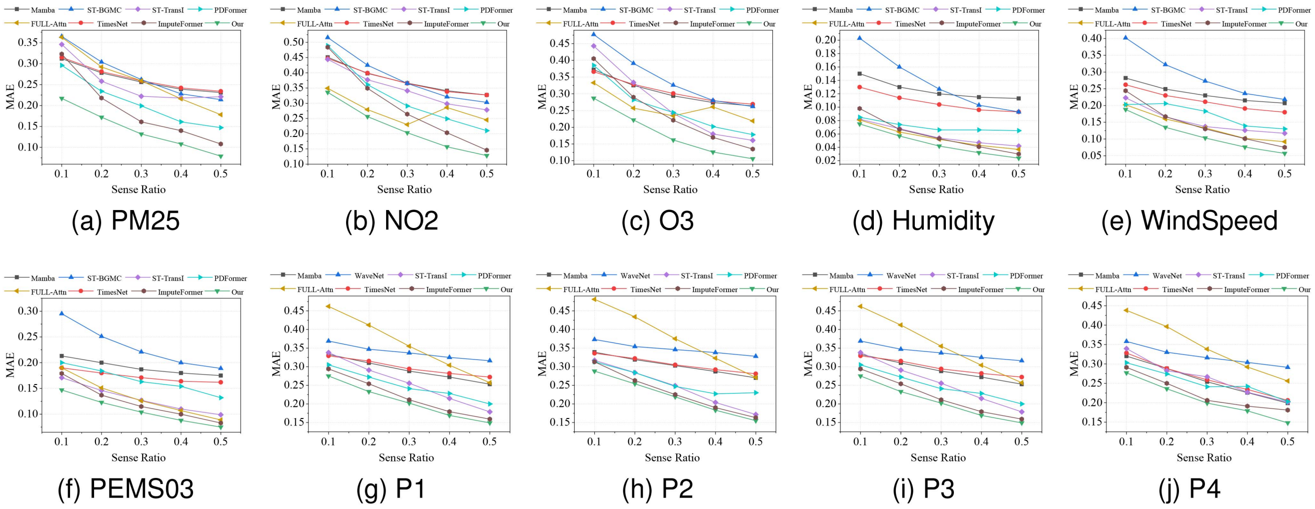
As shown in Fig. 9 for MSE and Fig. 10 for MAE, our model achieves the best performance across all datasets for both MSE and MAE metrics, with performance steadily improving as the finest-scale sense ratio increases. Upon closer analysis, our method exhibits a more significant advantage in handling high-variability datasets, such as the Air-Quality and Traffic datasets. For example, on the TaxiBJ dataset, as the finest-scale sense ratio increases, the reduction in both MSE and MAE for our model is more pronounced compared to single-scale methods, highlighting its superior capability in urban transportation scenarios characterized by dynamic and complex patterns. This



(a) PM25 (b) NO2 (c) O3 (d) Humidity (e) WindSpeed

(f) PEMS03 (g) P1 (h) P2 (i) P3 (j) P4

Fig. 9. Completion MSE performance under different sensed ratios.



(a) PM25 (b) NO2 (c) O3 (d) Humidity (e) WindSpeed

(f) PEMS03 (g) P1 (h) P2 (i) P3 (j) P4

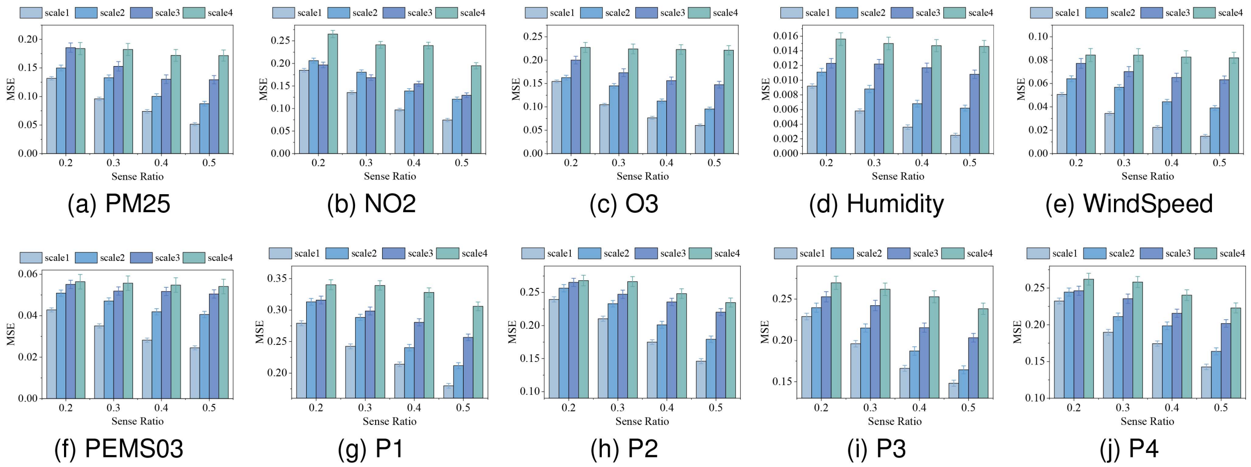
Fig. 10. Completion MAE performance under different sensed ratios.

is due to our model's ability to fully leverage the multi-scale nature of the data, making it more efficient in inferring data under highly variable conditions.

In contrast, single-scale models show notable limitations in capturing these complex spatio-temporal patterns, especially in highly dependent data scenarios. At the same time, although the FULL-Attn model also attempts to capture multi-scale relationships, its performance is relatively unstable. As the sense ratio increases, its effectiveness even decreases in some cases. This observed performance fluctuation and occasional degradation of FULL-Attn can be attributed to two main factors. First, its global attention mechanism, with quadratic complexity, is highly sensitive to variations and noise in sparse inputs, potentially struggling to discern critical signals from redundant information. Second, by attempting to model all-to-all relationships across all scales without explicit hierarchical guidance, FULL-Attn may inefficiently diffuse its attention, sometimes overemphasizing less relevant cross-scale interactions, which leads to poorer results compared to our structured pyramid-based approach.

3) Performance Under Varying Multi-Scale Sense Ratios: To validate the utilization of different scales, we conduct experiments to assess how varying sense ratios at different scales impacts completion performance. The experiments are divided into two phases: the first phase examines the impact of changing the sense ratio at a single scale on overall model performance, while the second phase focuses on the effect of simultaneous changes in sense ratios across two scales.

First, we adjust the sense ratio at one scale while keeping the others constant at 0.1. As shown in Fig. 11, the results across ten datasets reveal a clear improvement in the model's completion performance as the sense ratio at the targeted scale increases. This shows that our model effectively leverages data from different scales. Notably, while the total spatial coverage remains the same, the finer-scale data significantly enhance



(a) PM25 (b) NO2 (c) O3 (d) Humidity (e) WindSpeed

(f) PEMS03 (g) P1 (h) P2 (i) P3 (j) P4

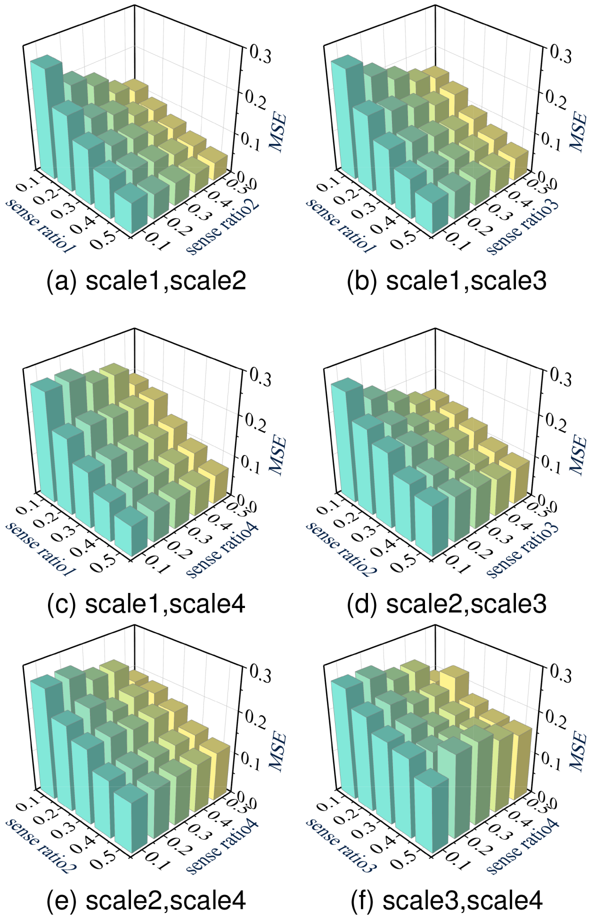
Fig. 11. Completion performance under single scale variation.

performance compared to coarser-scale data. Finer-scale data contribute not only to more substantial error reductions but also to more consistent and stable outputs, with fewer fluctuations. While this general trend holds, the grid-based aggregated flow data in TaxiBJ may exhibit slightly different sensitivities to scale variations, potentially due to the inherent smoothing effect of aggregation. These observations are consistent with real-world applications, where finer-grained data typically result in more accurate and representative inferences.

We further explore the impact on model completion performance as the sense ratios of two scales are increased. As shown in Fig. 12, the results indicate that as the sense ratios across multiple scales increase simultaneously, the overall model performance improves. However, an interesting phenomenon is observed when simultaneously increasing the sense ratios of the two coarsest scales, particularly at higher sense ratios or on datasets with a limited number of spatial points, where performance gains may diminish or show irregularities. This could be attributed to an increased likelihood of redundant information from overlapping fine-grained regions covered by these highly-sampled coarse scales, and the inherent uncertainty of aggregated coarse-scale data potentially becoming a limiting factor when finer-scale context is also sparse.

4) Performance Under Single-Scale Inputs: To evaluate the applicability of our model when provided with single-scale inputs, we conduct experiments across six real-world datasets. In this setting, only the finest-scale data is input into the model, while data from other scales are treated as unsensed (with a sense ratio of 0). The goal of this experiment is to determine whether multi-scale processing still contributes to performance improvements even when only single-scale data is available.

As shown in Fig. 13, our model outperforms most baseline methods across the majority of datasets and sensing ratios, particularly when the sense ratio is higher. This indicates that even with single-scale inputs, incorporating multi-scale processing can still improve performance. However, despite the superior performance in most cases, the results are not as strong as those obtained under multi-scale input scenarios. In some specific cases, the performance of our model is even worse than ST-TransI and TimesNet, suggesting that our method is better suited for multi-scale input scenarios. This also implies that multi-scale processing alone is not the sole reason for our model's superior results.



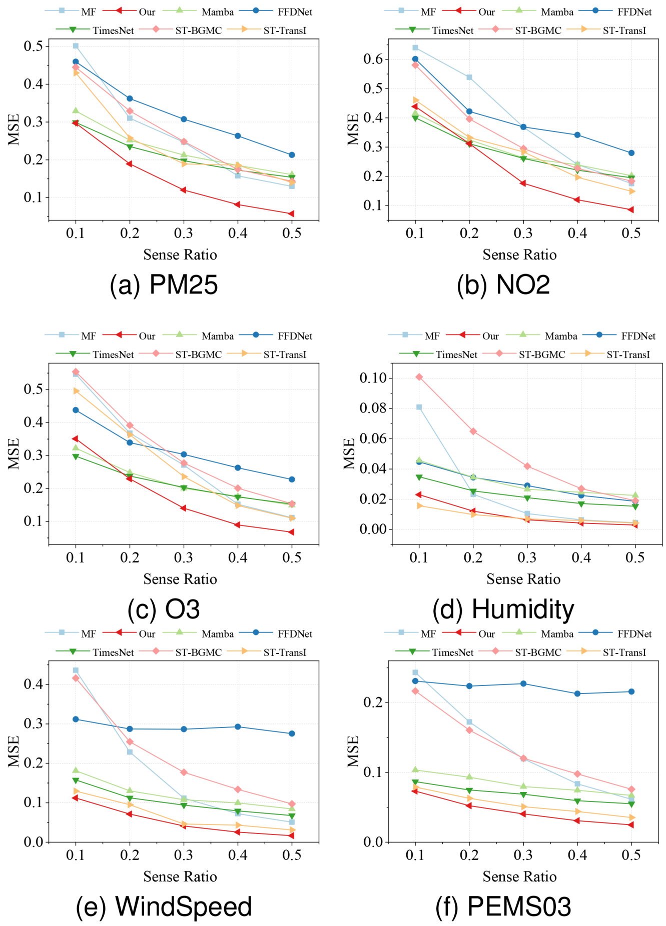
(a) scale1,scale2 (b) scale1,scale3

(c) scale1,scale4 (d) scale2,scale3

(e) scale2,scale4 (f) scale3,scale4

Fig. 12. Completion performance under dual scale variation.

Furthermore, as the sense ratio increases, the superiority of our model becomes more pronounced. With a higher sense



(a) PM25

(b) NO2

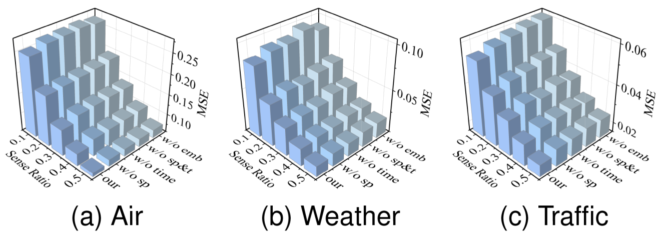
(c) O3

(d) Humidity

(e) WindSpeed

(f) PEMS03

Fig. 13. Performance under single-scale inputs.

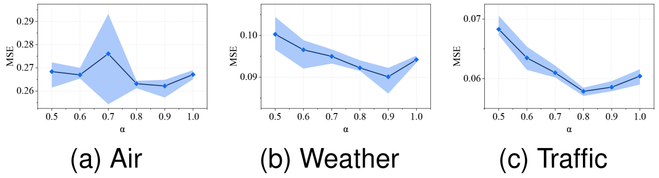


(a) Air

(b) Weather

(c) Traffic

Fig. 14. The effect of multi-scale embedding.



(a) Air

(b) Weather

(c) Traffic

Fig. 15. The effect of cross-scale constraint.

TABLE III
Performance of Different Pyramid Construction Methods

method	Air		Weather		Traffic	
	MSE	MAE	MSE	MAE	MSE	MAE
ST-PC	**0.2652**	**0.3358**	**0.0837**	**0.1876**	**0.0548**	**0.1470**
T-PC	0.2868	0.3487	0.0965	0.1983	0.0635	0.1581
S-PC	0.4576	0.4710	0.1128	0.2229	0.0693	0.1656

ratio, more available data allows the model to better capture multi-scale processes, thereby improving overall performance. In contrast, when the data is too sparse, the model struggles to fully leverage the multi-scale processes, resulting in poorer performance. This highlights the importance of data availability in the success of multi-scale frameworks and suggests that our model performs best when provided with sufficient data.

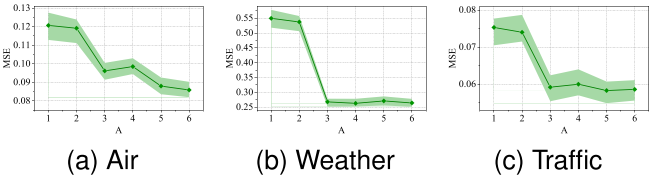
C. Ablation Study

In this subsection, we aim to validate the effectiveness of various components of our model, including the Multi-Scale Embedding, Cross-Scale Constraint, and ST-PC. To ensure a comprehensive analysis, in subsequent experiments, we use $\textbf{NO2}$ to represent the Air-Quality dataset, $\textbf{WindSpeed}$ for Weather dataset, and $\textbf{PEMS03}$ for Traffic dataset.

1) Multi-Scale Embedding: We design an ablation experiment to validate the effectiveness of the Multi-Scale Embedding Module. Specifically, we compare the performance of our model with and without different embedding components. As shown in Fig. 14, across all datasets and varying sense ratios, our full model consistently achieves the lowest MSE, demonstrating the importance of all embedding components. Notably, removing spatio-temporal embedding (“w/o sp&t”) leads to a significant performance degradation compared to the full model. Similarly, ablating either spatial (“w/o sp”) or temporal (“w/o time”) embedding components individually also results in poorer performance, though the extent varies by dataset, reflecting their differing sensitivities to spatial versus temporal context. Otherwise, the scale embedding component also has a big impact on the model, even more important than single spatial or temporal embedding component.

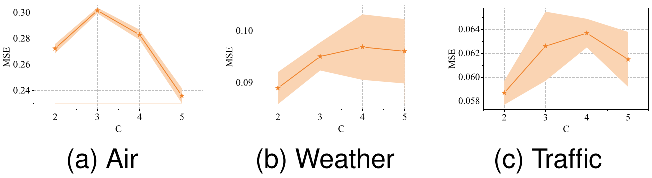
2) Cross-Scale Constraint: To assess the effectiveness of the Cross-Scale Constraint module, we repeat each experiment five times, calculate the average performance, and present the variability of the results through confidence intervals. As shown in Fig. 15, when the parameter α is appropriately selected, the Cross-Scale Constraint module significantly improves the model performance. However, if α is set too low, the module has an adverse effect, reducing the overall efficiency of the model. This suggests that careful tuning of α is critical to fully leverage the benefits of the Cross-Scale Constraint.

3) ST-PC: To validate the effectiveness of our ST-PC method, we conduct a comparative experiment among ST-PC, T-PC, and S-PC. Specifically, the T-PC method focuses solely on temporal multi-scale extraction, while the S-PC method concentrates only on spatial multi-scale extraction. As shown in Table III, our ST-PC method achieves the best performance. The S-PC method, which relies solely on the finest scale data and ignores other scales, performs poorly. Although the T-PC method leverages coarser scale data, its performance is constrained by the sparsity of the inputs. Our ST-PC method, which uniquely



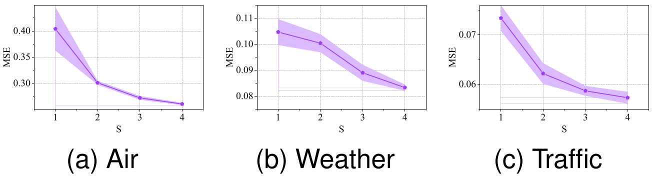
(a) Air (b) Weather (c) Traffic

Fig. 16. The completion effect under different A settings.



(a) Air (b) Weather (c) Traffic

Fig. 17. The completion effect under different C settings.



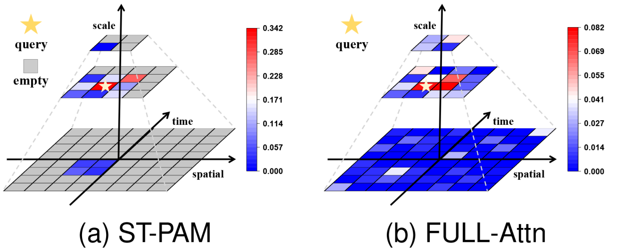
(a) Air (b) Weather (c) Traffic

Fig. 18. The completion effect under different S settings.

integrates both the original and newly constructed data, achieves the most effective results. These observations ultimately suggest that under sparse collection scenarios, utilizing as much available information as possible may lead to significant performance improvements.

4) Multi-Scale Parameter Sensitivity: To provide deeper insights into different multi-scale scenarios, we conducted sensitivity analyses on three key hyperparameters: A, C, S. For parameter A, as shown in Fig. 16, when $A < 3$, it does not satisfy (34), leading to lack of global receptive field, potentially limiting performance. However, when $A \geq 3$, model performance significantly improves, while further increasing A yields diminishing returns.

For parameter C, which describes the aggregation relationship between adjacent scales, a larger C implies that each coarse-scale unit aggregates more fine-scale information, but this also increases the uncertainty and level of abstraction of the resulting coarse-scale features. As shown in Fig. 17, effectively leveraging this highly aggregated and uncertain information would require stronger model inference capabilities; thus, a moderate value like $C = 2$ offers a better balance between information aggregation and detail preservation. Regarding the total number of scales S, as shown in Fig. 18, performance consistently improves as S increases within our tested range. This is because a larger S provides the model with a richer hierarchy and a greater volume of multi-scale information to learn from. Crucially, while the specific choices of S and C do lead to performance variations, these fluctuations are generally acceptable. Our model consistently outperforms the state-of-the-art single-scale baselines across all configurations, which validates our framework's robustness and its ability to effectively adapt to complex multi-scale sensing environments.

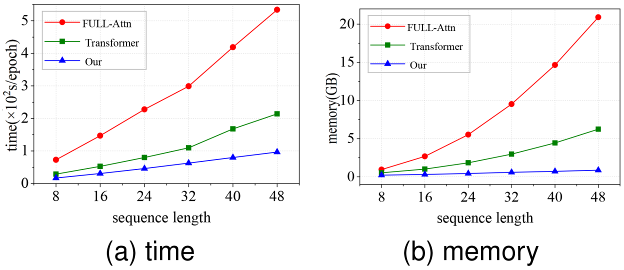


(a) ST-PAM (b) FULL-Attn

Fig. 19. Attention score visualization of ST-PAM and Full-Attn.

TABLE IV
COMPUTING OVERHEAD FOR MAIN METHODS

method	**Transformer**	**FULL-Attn**	**Our**
Q-K pairs	17039360	30642176	1262208
memory cost	2GB	4GB	<1GB



(a) time (b) memory

Fig. 20. The time and memory consumption.

D. Attention Analysis

As shown in Fig. 19, we compare the attention mechanisms by visualizing their scores for a specific node. Our ST-PAM selectively focuses on a small subset of the most relevant nodes, efficiently capturing key spatio-temporal correlations. In contrast, Full-Attention distributes its attention more broadly, including many less relevant nodes. Notably, ST-PAM identifies the same critical nodes as Full-Attention but avoids wasting computational resources on unimportant ones, thus achieving a more efficient attention allocation.

E. Computing Overhead

We compare the computing overhead of our model with the Transformer model and the Full attention-based multi-scale model from two perspectives: the number of query-key dot products (Q-K pairs), and memory cost. As shown in Table IV, our model demonstrates a significant reduction in both computational complexity and memory.

Furthermore, we conducted experiments on a single NVIDIA GeForce RTX 3090 GPU, evaluating the runtime and memory overhead across varying input sequence lengths. As shown in Fig. 20, our model not only maintains a near-linear scaling in both time and memory consumption, but also exhibits a remarkably competitive, even lower computational cost than single-scale methods, particularly for longer sequences.

VII. Conclusion

In conclusion, this paper presents a groundbreaking multi-scale spatio-temporal data completion framework for Sparse CrowdSensing using a Pyramid-Attention based approach. The framework addresses the challenges of data sparsity and multi-scalability, integrating innovative components like the Multi-scale Embedding Layer and the Spatial-Temporal Pyramid Construction Module (ST-PC). The proposed Spatial-Temporal Pyramid Attention Mechanism (ST-PAM) efficiently extracts multi-scale correlations, maintaining linear computational complexity. Our work presents a new perspective for handling spatio-temporal data in Sparse Crowdsensing, offering valuable insights into the potential of multi-scale processing. In future work, beyond optimizing the collection of multi-scale data for effective utilization, we also plan to explore the integration of domain-specific prior knowledge and explicit semantic relationships to further enhance model performance and interpretability in targeted application scenarios, potentially refining the balance between general applicability and specialized accuracy. This could further enhance the capabilities of multi-scale models in practical applications.

References

[1] C. M. Gussen, P. S. Diniz, M. L. Campos, W. A. Martins, F. M. Costa, and J. N. Gois, “A survey of underwater wireless communication technologies,” *J. Commun. Inf. Syst.*, vol. 31, no. 1, pp. 242–255, 2016.

[2] P. P. Parikh, M. G. Kanabar, and T. S. Sidhu, “Opportunities and challenges of wireless communication technologies for smart grid applications,” in *Proc. IEEE PES Gen. Meeting*, 2010, pp. 1–7.

[3] R. K. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: Current state and future challenges,” *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.

[4] F. Calabrese, G. Di Lorenzo, and C. Ratti, “Human mobility prediction based on individual and collective geographical preferences,” in *Proc. 13th Int. IEEE Conf. Intell. Transp.*, 2010, pp. 312–317.

[5] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, “Sensor-based activity recognition,” *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 790–808, Nov. 2012.

[6] A. Crooks, A. Croitoru, A. Stefanidis, and J. Radzikowski, “#Earthquake: Twitter as a distributed sensor system,” *Trans. GIS*, vol. 17, no. 1, pp. 124–147, 2013.

[7] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M’hamed, “Sparse mobile crowdsensing: Challenges and opportunities,” *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 161–167, Jul. 2016.

[8] S. Zhao, G. Qi, T. He, J. Chen, Z. Liu, and K. Wei, “A survey of sparse mobile crowdsensing: Developments and opportunities,” *IEEE Open J. Comput. Soc.*, vol. 3, no. 1, pp. 73–85, May. 2022.

[9] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[10] E. Candes and B. Recht, “Exact matrix completion via convex optimization,” *Commun. ACM*, vol. 55, no. 6, pp. 111–119, 2012.

[11] Y. Wu, D. Zhuang, M. Lei, A. Labbe, and L. Sun, “Spatial aggregation and temporal convolution networks for real-time kriging,” 2021, *arXiv:2109.12144*.

[12] E. Wang et al., “Deep learning-enabled sparse industrial crowdsensing and prediction,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6170–6181, Sep. 2021.

[13] E. Wang et al., “A new data completion perspective on sparse crowdsensing: Spatiotemporal evolutionary inference approach,” *IEEE Trans. Mobile Comput.*, vol. 24, no. 3, pp. 1357–1371, Mar. 2025.

[14] K. Ouyang et al., “Fine-grained urban flow inference,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 6, pp. 2755–2770, Jun. 2022.

[15] F. Zhou, X. Jing, L. Li, and T. Zhong, “Inferring high-resolutional urban flow with Internet of Mobile Things,” in *Proc. 2021 IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 7948–7952.

[16] Y. Zhang et al., “Skilful nowcasting of extreme precipitation with NowcastNet,” *Nature*, vol. 619, no. 7970, pp. 526–532, 2023.

[17] H. Al-Mekhlafi and S. Liu, “Single image super-resolution: A comprehensive review and recent insight,” *Front. Comput. Sci.*, vol. 18, no. 1, 2024, Art. no. 181702.

[18] S. Liu et al., “PyraFormer: Low-complexity pyramidal attention for long-range time series modeling and forecasting,” in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–20.

[19] L. Chen et al., “Multi-scale adaptive graph neural network for multivariate time series forecasting,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 10, pp. 10748–10761, Oct. 2023.

[20] S. Wang et al., “TimeMixer: Decomposable multiscale mixing for time series forecasting,” in *Proc. 12th Int. Conf. Learn. Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=7oLshfEIC2

[21] Y. Yuan, Y. Zhang, B. Wang, Y. Peng, Y. Hu, and B. Yin, “STGAN: Spatio-temporal generative adversarial network for traffic data imputation,” *IEEE Trans. Big Data*, vol. 9, no. 1, pp. 200–211, Feb. 2023.

[22] J. Li, C. Gsaxner, A. Pepe, D. Schmalstieg, J. Kleesiek, and J. Egger, “Sparse convolutional neural network for high-resolution skull shape completion and shape super-resolution,” *Sci. Rep.*, vol. 13, no. 1, 2023, Art. no. 20229.

[23] E. Wang, W. Liu, W. Liu, C. Xiang, B. Yang, and Y. Yang, “Spatiotemporal transformer for data inference and long prediction in sparse mobile crowdsensing,” in *Proc. IEEE Int. Conf. Comput. Commun.*, 2023, pp. 1–10.

[24] E. Wang, M. Zhang, B. Yang, Y. Xu, Z. Song, and Y. Yang, “Few-shot data completion for new tasks in sparse crowdsensing,” in *Proc. IEEE Conf. Comput. Commun.*, 2024, pp. 1831–1840.

[25] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Cambridge, MA, USA: MIT Press, 2010.

[26] R. Zhang, “Making convolutional networks shift-invariant again,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7324–7334.

[27] A. Vaswani et al., “Attention is all you need,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[28] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020, *arXiv: 2010.11929*.

[29] S. Ren, D. Zhou, S. He, J. Feng, and X. Wang, “Shunted self-attention via multi-scale token aggregation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10853–10862.

[30] Q. Fan, H. Huang, M. Chen, H. Liu, and R. He, “RMT: Retentive networks meet vision transformers,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 5641–5651.

[31] A. Crivellari, H. Wei, C. Wei, and Y. Shi, “Super-resolution GANs for upscaling unplanned urban settlements from remote sensing satellite imagery–the case of chinese urban village detection,” *Int. J. Digit. Earth*, vol. 16, no. 1, pp. 2623–2643, 2023.

[32] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16000–16009.

[33] H. Zhou et al., “Informer: Beyond efficient transformer for long sequence time-series forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11106–11115.

[34] N. Cressie and C. K. Wikle, *Statistics for Spatio-Temporal Data*. Hoboken, NJ, USA: Wiley, 2011.

[35] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[36] Z. Wen, W. Yin, and Y. Zhang, “Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm,” *Math. Program. Computation*, vol. 4, no. 4, pp. 333–361, 2012.

[37] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2016, *arXiv:1609.02907*.

[38] K. Zhang, W. Zuo, and L. Zhang, “FFDNet: Toward a fast and flexible solution for CNN-based image denoising,” *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[39] A. Gu and T. Dao, “Mamba: Linear-time sequence modeling with selective state spaces,” 2023, *arXiv:2312.00752*.

[40] W. Du, D. Côté, and Y. Liu, “SAITS: Self-attention-based imputation for time series,” *Expert Syst. Appl.*, vol. 219, 2023, Art. no. 119619.

[41] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, “TimesNet: Temporal 2D-variation modeling for general time series analysis,” in *Proc. 11th Int. Conf. Learn. Representations*, 2023, pp. 1–23.

[42] K. E. Smith and A. O. Smith, "Conditional GAN for timeseries generation," 2020, *arXiv: 2006.16477*.

[43] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 1907–1913.

[44] W. Liu, Y. Yang, E. Wang, and J. Wu, "Fine-grained urban prediction via sparse mobile crowdsensing," in *Proc. IEEE 17th Int. Conf. Mobile Ad Hoc Sensor Syst.*, 2020, pp. 265–273.

[45] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 4365–4373.

[46] T. Nie, G. Qin, W. Ma, Y. Mei, and J. Sun, "ImputeFormer: Low rankness-induced transformers for generalizable spatiotemporal imputation," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2024, pp. 2260–2271.

Wenbin Liu received the BS degree in physics and the PhD degree in computer science and technology from Jilin University, China, in 2012 and 2020, where he is currently an associate professor with the College of Computer Science and Technology. He is currently also a postdoctoral researcher with China Telecom. His research interests include mobile crowdsensing, spatio-temporal crowdsourcing, and ubiquitous computing.

Hao Du received the BE degree in software engineering from Inner Mongolia University, Hohhot, China, in 2022. He enrolled in a combined master's and PhD program in computer science and technology from Jilin University, Changchun, China, in 2022, where he is currently working toward the PhD degree. His current research focuses on mobile crowdsensing, spatio-temporal data processing, and multi-scale data processing.

En Wang (Member, IEEE) received the BE degree in software engineering from Jilin University, Changchun, in 2011, and the ME and PhD degrees in computer science and technology from Jilin University, Changchun, in 2013 and 2016, respectively. He was also a joint PhD student with the Department of Computer and Information Science, Temple University. He is currently a professor with the Department of Computer Science and Technology, Jilin University, Changchun. His current research focuses on mobile computing, crowd intelligence, and data mining.

Jiajian Lv received the BE degree in software engineering from Jilin University, Changchun, China, in 2024. He is currently working toward the ME degree in computer science and technology with Jilin University, Changchun, China. His current research focuses on data mining, mobile computing, and spatio-temporal data processing.

Weiting Liu received the BE degree in software engineering and the ME degree in computer science and technology from Jilin University, Changchun, China, in 2020 and 2023. He is currently working toward the PhD degree in computer science and technology with Jilin University. His current research focuses on mobile crowdsensing, spatio-temporal data processing, and spatio-temporal dynamics.

Bo Yang is currently a professor with the College of Computer Science and Technology, Jilin University. He is also the director with the Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, China. His current research interests include the areas of data mining, complex network analysis, self-organized and self-adaptive multi-agent systems, with applications to knowledge engineering and intelligent health informatics.

Jie Wu (Fellow, IEEE) is the director with the Center for Networked Computing and Laura H.Carnell professor with Temple University. He also serves as the director of International Affairs with the College of Science and Technology. He served as chair with the Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and associate vice provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director with the National Science Foundation and was a distinguished professor with Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. He was an IEEE Computer Society distinguished visitor and ACM distinguished speaker. He is a China Computer Federation (CCF) distinguished speaker. Currently, he is on leaving working as a scientist with China Telecom.