Spatiotemporal Non-Aligned Data Inference for Sparse Mobile CrowdSensing

Hao Du, Wenbin Liu, *Member, IEEE*, En Wang, *Member, IEEE*, Zhaoyang Sun, *Graduate Student Member, IEEE*, Chenyang Fu, Bo Yang, and Jie Wu, *Fellow, IEEE*

***Abstract*—Sparse Mobile CrowdSensing is a practical paradigm for data collection, where mobile users report partial measurements and the platform infers the missing values by exploiting spatiotemporal correlations. Most existing methods assume uniform spatial scales and aggregate observations into fixed time slots. In practice, however, heterogeneous devices and asynchronous reporting lead to observations that vary in spatial coverage and arrive at irregular timestamps. Spatially, coarse-scale measurements often cannot be perfectly decomposed into fine-scale grids due to offset or partial overlaps; temporally, continuous irregular observations may misalign with the desired target reporting moments. These spatial and temporal misalignments jointly make data inference substantially more challenging. To tackle this problem, we propose ST-NAI, a unified framework for spatiotemporal non-aligned data inference. ST-NAI models continuous temporal dynamics with Time-Mamba by fusing physical time intervals, and resolves geometric mismatches via a scale-adjusted resampling strategy. We further enhance spatiotemporal integration with ST-Mamba and Cross-Mamba to capture global context with linear complexity. Experiments on six real-world datasets demonstrate the effectiveness of the proposed framework in handling data inference challenges.**

***Index Terms*—Sparse mobile crowdsensing, multi-scale processing, spatiotemporal data, spatiotemporal non-aligned.**

I. Introduction

SPARSE Mobile CrowdSensing (Sparse MCS) [1] has emerged as a promising and cost-effective method for data collection in diverse real-world scenarios. Sparse MCS recruits participants equipped with mobile devices, such as smartphones, vehicles, or drones, to collect partial data. Then, data inference, the core of Sparse MCS, utilizes spatiotemporal correlations to reconstruct complete data. This ensures data completeness while reducing collection costs, thereby enhancing Quality of Service (QoS). Currently, Sparse MCS has demonstrated its effectiveness in traffic management [2], environmental monitoring [3], [4], and disaster relief [5].

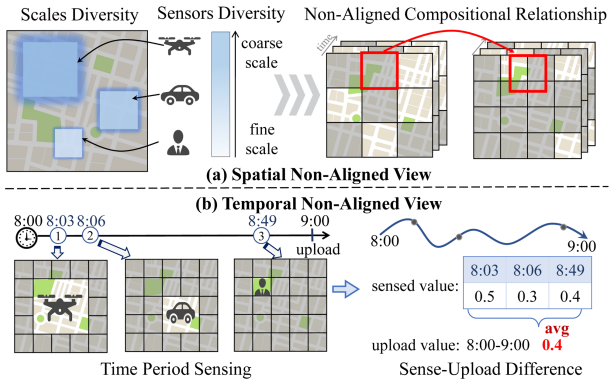


Fig. 1. The Spatiotemporal Non-Aligned Scenarios.

However, the complexities of data heterogeneity in both temporal and spatial dimensions have been significantly neglected by previous research. Existing methods in Sparse MCS, such as compressive sensing [6], matrix completion [7], and deep learning-based techniques [8], [9], typically simplify the problem by assuming that the scales of collected data are identical or similar, and that observations are aggregated into fixed, synchronized time slots. Nevertheless, in real-world scenarios, the heterogeneity of sensing devices combined with the stochastic mobility and asynchronous reporting behaviors of participants inevitably result in highly irregular multi-scale data collections.

Specifically, from the spatial perspective, the diversity of device specifications results in heterogeneous sensing capabilities and spatial coverages, which naturally leads to data collection at different spatial scales. As shown in Fig. 1(a), consider a scenario in which three users equipped with different devices (a smartphone, a vehicle, and a drone) collect traffic flow data, illustrating a typical multi-scale setting. In practice, different scales are represented by grids with different cell sizes: finer scales use smaller cells to capture localized variations, while coarser scales use larger cells to summarize a broader area. There often exists a compositional relationship across scales, and this relationship stems from the intrinsic physical and statistical properties of

Received 22 December 2025; revised 19 March 2026; accepted 25 April 2026. Date of publication 30 April 2026; date of current version 6 August 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 92567204, Grant 62272193, and Grant 62472194, in part by the Fundamental Research Funds for the Central Universities, and Jilin Science and Technology Research Project under Grant 20260203130SF, and in part by the Graduate Innovation Fund of Jilin University under Grant 2026CX206. An earlier version of this paper was presented in part at the 33rd IEEE/ACM International Symposium on Quality of Service 2025 (IWQoS) [DOI: 10.1109/IWQoS65803.2025.11143411]. Recommended for acceptance by D. Yang. *(Corresponding authors: Wenbin Liu; En Wang; Jie Wu.)*

Hao Du, Wenbin Liu, En Wang, Zhaoyang Sun, Chenyang Fu, 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: duhao22@mails.jlu.edu.cn; liuwenbin@jlu.edu.cn; wangen@jlu.edu.cn; zys25@mails.jlu.edu.cn; fucy25@mails.jlu.edu.cn; ybo@jlu.edu.cn).

Jie Wu is with 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.2026.3689085

1536-1233 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

the sensed variables. For example, environmental data such as PM2.5 or humidity typically represent spatial averages over a region, while traffic flow reflects cumulative counts over finer spatial units. Ideally, such mappings are straightforward when the coarse and fine grids are perfectly aligned. However, real-world sensing footprints and grid partitions are often offset or only partially overlap, so a coarse cell may not be exactly decomposed into fine cells, leading to spatial non-alignment.

From the temporal perspective, stochastic mobility and asynchronous reporting behaviors lead to continuous and irregular data streams. As shown in Fig. 1(b), unlike static sensors configured with fixed sampling frequencies, mobile participants upload data based on their dynamic trajectories and random availability. Consequently, data points appear at arbitrary timestamps (e.g., 8:03, 8:49) rather than synchronized intervals. Standard approaches typically aggregate these distinct observations into a single value to represent a fixed time slot. However, this operation masks the true dynamic fluctuation trajectory within the hour, leading to temporal information loss. Furthermore, the aggregated data inherently misalign with the precise target reporting moment (e.g., 9:00), posing a significant challenge for accurate inference. These geometric mismatches in space and dynamic irregularities in time cause a complex heterogeneity that fundamentally undermines the effectiveness of traditional methods. Therefore, developing methods for the spatiotemporal non-aligned data inference in Sparse MCS is critical.

To develop an effective spatiotemporal non-aligned data inference method for Sparse MCS, three challenges arise:

1) **How to address temporal non-alignment caused by irregular temporal sampling?** To avoid the misalignment caused by averaging irregular observations, a direct strategy is to perform inference at the finest temporal granularity. However, taking the observations at 8:03, 8:06, and 8:49 with a target time at 9:00 as an example, aligning them to the minute-level granularity results in a temporal sequence of length 60, yet containing only three valid data points. Beyond extreme sparsity, the extensive unobserved time steps lack ground truth supervision, making it fundamentally difficult for the model to learn the continuous state transitions. Alternatively, directly feeding the valid observations into standard sequence models (i.e., t, $t+1$, $t+2$) avoids this sparsity but compromises temporal sensitivity. By processing these points as equidistant steps, the model treats the short interval between 8:03 and 8:06 (3 mins) identically to the significant duration between 8:06 and 8:49 (43 mins). This insensitivity to varying physical time gaps makes it effectively impossible to capture the true evolutionary dynamics required to infer the state at the non-aligned target moment.
2) **How to effectively handle multi-scale data with spatially non-aligned compositional relationships?** Recently, a few data mining works, such as Pyraformer [10] and Timemixer [11], have explored the construction and handling of multi-scale data. These studies typically rely on artificially constructed multi-scale structures, where coarse scales are generated via integer-based downsampling. This results in aligned multi-scale data with consistent hierarchical boundaries, which can be easily processed using tree-like or convolutional frameworks. However, in practical MCS scenarios, the collected coarse-scale data points often do not decompose perfectly into multiple fine-scale data points (e.g., the mapping between 4×4 and 3×3 grids). Standard methods, limited by integer strides, cannot model such geometric relationships. This leads to spatial feature mismatches between data at different scales, posing significant challenges for processing multi-scale data.
3) **How to construct a lightweight and effective model for spatiotemporal integration?** After addressing the non-alignment issues in temporal and spatial dimensions, it is critical to integrate the spatiotemporal information. Traditional machine learning methods, despite their low computational costs, primarily address linear spatiotemporal relationships and perform poorly in adapting to non-aligned data processing. Conversely, deep learning methods like Transformers [12] are highly effective and adaptable but lead to significant computational costs. In this context, Mamba [13], an effective and lightweight parallel processing architecture, has attracted our attention. However, Mamba is inherently designed for 1D sequential modeling. This design creates a gap when processing 2D spatial structures, thereby limiting its applicability in handling spatiotemporal data.

To address these challenges, we propose a novel Spatio-temporal Non-Aligned Data Inference framework for Sparse MCS, named ST-NAI.

1) **To handle temporal non-alignment**, we propose Time-Mamba, which explicitly fuses physical time intervals into the Mamba model. It accurately models continuous dynamics, effectively bridging the gap between irregular observations and target moments.
2) **To handle spatial non-alignment**, we design a Scale-Adjusted Resampling strategy based on least common multiple scaling. It aligns multi-scale structures, effectively resolving spatial feature mismatches.
3) **To achieve efficient spatiotemporal integration**, we propose ST-Mamba and Cross-Mamba. By introducing spatial information interaction and multi-directional correlation capture, they effectively achieve sensitivity to spatiotemporal information. Crucially, both modules ensure linear computational complexity, achieving lightweight and effective processing.

Our work has the following contributions:

- We investigate the problem of spatiotemporal non-aligned data inference in Sparse Mobile CrowdSensing. We propose a novel framework named ST-NAI, which unifies continuous temporal modeling and non-aligned spatial processing to achieve comprehensive data inference.
- For the temporal dimension, we propose Time-Mamba to model continuous temporal dynamics by explicitly utilizing physical time intervals, effectively bridging the gap between irregular observations and target moments. For the spatial dimension, we design a Scale-Adjusted Resampling

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

strategy based on least common multiple scaling to resolve geometric non-alignment.
- We develop ST-Mamba and Cross-Mamba for efficient spatiotemporal integration. By introducing spatial information interaction and multi-directional correlation capture, they achieve high spatiotemporal sensitivity while maintaining linear computational complexity.
- We evaluate the proposed method on six real-world datasets, where it consistently outperforms competitive baselines with an average error reduction of approximately 10–20% across different evaluation metrics. These results demonstrate its effectiveness in addressing the spatiotemporal non-aligned data inference problem.

The remainder of this paper is organized as follows. After reviewing the related works in Section II, we introduce the system model and preliminaries in Section III. Then, the spatiotemporal non-aligned data inference method is proposed in Section IV, followed by the theoretical analysis in Section V. Finally, we evaluate the performance in Section VI and conclude this paper in Section VIII.

II. Related Work

A. Sparse Mobile CrowdSensing

Sparse MCS [1] is a practical solution for data collection, which recruits users to sense partial data and leverages spatiotemporal correlations to infer the rest. With recent technological advancements, deep learning-based methods have become the *de facto* choice in Sparse MCS. Marchang et al. [14] explored a KNN-based inference framework exploiting spatiotemporal correlations, while Yang et al. [15] proposed a graph attention-based reconstruction model to capture complex spatiotemporal dependencies. Wang et al. [7] further introduced a deep learning-enabled approach, and Liu et al. [8] developed a neural network-based fine-grained completion and prediction framework. Furthermore, Wang et al. [9] proposed a spatiotemporal Transformer for data inference and long-term prediction. Beyond Sparse MCS, attention-based imputation models have also been studied: Du et al. [16] proposed SAITS with diagonally-masked self-attention, and Nie et al. [17] introduced ImputeFormer by integrating low-rank priors with Transformers. However, most existing methods assume identical or similar data scales, which conflicts with heterogeneous sensing devices in practice; addressing this discrepancy in multi-scale contexts remains a critical challenge.

B. Multi-Scale Model

In recent years, many studies have focused on the analysis of multi-scale features in spatiotemporal data. Liu et al. [10] introduced Pyraformer to enhance prediction accuracy by capturing scale correlations with a pyramid attention mechanism. Similarly, Wang et al. [11] introduced TimeMixer, employing a decomposable multi-scale mixing mechanism to integrate information from various scales. Chen et al. [18] constructed the Multi-Scale Adaptive Graph Neural Network, improving feature extraction and integration by learning multi-scale graph topologies. Additionally, Zhang et al. [19] presented CorrFormer, which uses tree structures to reduce computational complexity while maintaining prediction performance. Despite these advancements, the diversity of sensing devices contributes to challenges with non-aligned multi-scale inputs, underscoring the pressing need for further research in this area.

C. Mamba and Its Variants

Mamba [13] has been proposed as an efficient architecture for handling long-range dependencies in sequential data. Recently, many studies have focused on the application and extension of Mamba. For example, Liu et al. [20] introduced VMamba, which employs a visual state space model for efficient visual representation learning. This method incorporates a cross-scan module to traverse the spatial domain, transforming non-causal visual images into ordered block sequences for processing. Zhu et al. [21] developed Vision Mamba, which enhances visual representation learning efficiency by randomly shuffling visual images and utilizing a bidirectional state space model. Additionally, Li et al. [22] proposed STG-Mamba, which leverages a selective state space model for spatiotemporal graph learning. Despite these advancements, since Mamba is designed for sequential tasks, its handling of spatial information is limited. Current adaptations struggle with non-intuitive sequential traversals. Enhancing Mamba's spatial sensitivity remains a critical challenge.

D. Continuous Temporal Modeling

Modeling continuous dynamics from irregularly sampled time series is a fundamental challenge in machine learning. For instance, Zhu et al. [23] proposed Time-LSTM, which equips LSTM units with specialized time gates to model the intervals between user actions. However, a significant paradigm shift occurred with the introduction of Neural Ordinary Differential Equations (Neural ODEs) [24], which model the evolution of hidden states as continuous functions of time using numerical solvers. Latent ODE-LSTM [25] combines LSTM encoders with ODE decoders and utilizes gradient clipping to mitigate training instability. Subsequently, ODE-RSSM [26] integrates ODEs into recurrent state space models to identify nonlinear and stochastic systems from irregular data. More recently, DIFFODE [27] introduces a differentiable hidden-state framework with attention mechanisms to better map irregular observations into continuous spaces. Despite their theoretical elegance, these ODE-based methods rely on expensive iterative solvers, resulting in high computational overhead and over-smoothed latent trajectories. Consequently, they are inadequate for effectively handling the complex MCS sensing environments.

III. System Model and Preliminaries

A. System Model

We consider a comprehensive Sparse Mobile CrowdSensing scenario where a platform recruits participants equipped with heterogeneous devices to sense partial data at different locations and time instants. Unlike conventional approaches that aggregate measurements into fixed discrete time slots (e.g., hourly

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

averages), we model data collection as a time-continuous and irregular process. The platform aggregates sparse, multi-scale, and temporally irregular observations and aims to recover the sensing maps at specified target times. For convenience, we discretize the sensing region into a grid of $H \times W$ subareas at the finest spatial resolution.

Time: Let $\mathcal{T}_{obs} = \{\tau_1, \tau_2, \ldots, \tau_L\}$ denote the irregular observation timestamps, where τ_k is the recording time of the k-th observation. We further define a set of discrete target timestamps $\mathcal{T}_{target} = \{t_1, t_2, \ldots, t_T\}$, at which the complete sensing maps are required.

Scale: We categorize the sensing spatial coverage into S scales. The scale factor at level s is denoted as $g^{(s)}$, which represents the magnification factor relative to the finest scale.[1] A scale of $g^{(s)}$ means that each cell at scale s covers an area of $g^{(s)} \times g^{(s)}$ cells at the finest scale.[2] Accordingly, the grid size at scale s is denoted as follows:

$$H^{(s)} = H/g^{(s)}, \quad W^{(s)} = W/g^{(s)}, \quad s = 1, 2, \ldots, S. \tag{1}$$

Data: For the k-th observation at timestamp τ_k, the sensed value at location (h, w) on scale s is denoted as $x^{(s)}_{k,h,w}$, and the corresponding ground truth is $y^{(s)}_{k,h,w}$. Let $\mathbf{Y}^{(s)} \in \mathbb{R}^{L \times H^{(s)} \times W^{(s)}}$ collect all ground-truth values at scale s. We define an observation mask $\mathbf{M}^{(s)} \in \{0, 1\}^{L \times H^{(s)} \times W^{(s)}}$, where $m^{(s)}_{k,h,w} = 1$ indicates an observed entry and $m^{(s)}_{k,h,w} = 0$ otherwise.[3] The observed data at scale s is denoted as follows:

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

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

Corresponding to the target timestamps $\mathcal{T}_{target}$, we denote the complete sensing maps as $\mathbf{Z}$, where $\mathbf{Z}^{(s)} \in \mathbb{R}^{T \times H^{(s)} \times W^{(s)}}$ is the ground truth at scale s, and $\mathbf{Z} = \{\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \ldots, \mathbf{Z}^{(S)}\}$.

Inference: Given $\mathbf{X}$, the proposed inference algorithm $\mathcal{I}(\cdot)$ outputs the reconstructed maps at target timestamps:

$$\hat{\mathbf{Z}} = \mathcal{I}(\mathbf{X}). \tag{3}$$

B. Problem Formulation

Problem [Non-Aligned Multi-Scale Data Inference]: Given sparse multi-scale observations $\mathbf{X}$ associated with irregular timestamps $\mathcal{T}_{obs}$, we aim to reconstruct complete sensing maps $\hat{\mathbf{Z}}$ at the target timestamps $\mathcal{T}_{target}$. In many practical applications, downstream tasks primarily require the highest-resolution (finest-scale) map, while coarser-scale maps can often be obtained via aggregation from the finest scale. Therefore, we define the objective on the finest scale:

$$\min \ \delta\left(\hat{\mathbf{Z}}^{(1)}, \mathbf{Z}^{(1)}\right), \tag{4}$$

$$\text{s.t.} \quad \hat{\mathbf{Z}} = \mathcal{I}(\mathbf{X}), \tag{5}$$

where $\delta(\cdot, \cdot)$ measures the discrepancy between the inferred result and the ground truth at the finest scale.

C. Workflow

Our workflow is illustrated in Fig. 2. Consider a real-world scenario where the platform publishes an urban sensing task and recruits users equipped with diverse devices, e.g., smartphones (fine scale), vehicles (mid scale), and drones (coarse scale). Instead of collecting data in fixed time slots, these users sense and report at continuous, irregular timestamps (e.g., 8:03, 8:06, 8:49) along their dynamic trajectories. The platform aggregates these sparse multi-scale snapshots into an observation sequence. The inference model then exploits the multi-scale observations to learn cross-scale representations and complete the data at the target timestamps. In particular, we produce the final high-resolution inference at the finest scale for downstream applications, while the coarser-scale outputs mainly serve as auxiliary representations and can be optionally obtained through the same inference process.

D. Preliminaries on State Space Models

1) State Space Models: State Space Models (SSM) [28] serve as a foundational framework for modeling the evolution of internal states over time in continuous systems. The linear time-continuous system maps a 1-D input sequence $x(t) \in \mathbb{R}$ to an output $y(t) \in \mathbb{R}$ through an implicit latent state $h(t) \in \mathbb{R}^N$, which can be expressed as:

$$h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t), \ y(t) = \mathbf{C}h(t), \tag{6}$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ represents the evolution parameter, $\mathbf{B} \in \mathbb{R}^{N \times 1}$ and $\mathbf{C} \in \mathbb{R}^{1 \times N}$ represent the projection parameters.

To apply this system to discrete sampled data in real-world scenarios, it is discretized by the Zero-Order Hold (ZOH) method. This transformation converts the continuous parameters $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ into discrete counterparts:

$$\bar{\mathbf{A}} = \exp(\Delta\mathbf{A}), \bar{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B}, \tag{7}$$

$$h_t = \bar{\mathbf{A}}h_{t-1} + \bar{\mathbf{B}}x_t, y_t = \mathbf{C}h_t, \tag{8}$$

where Δ represents the sampling time interval.

2) Mamba: Traditional SSMs are designed for linear time-invariant scenarios; however, many real-world scenarios are nonlinear time-variant. To overcome this limitation, Mamba [13] introduces an input-dependent selection mechanism that enables effective information filtering from the input. Specifically, the parameters $\mathbf{B} \in \mathbb{R}^{B \times L \times D}$, $\mathbf{C} \in \mathbb{R}^{B \times L \times D}$ and $\Delta \in \mathbb{R}^{B \times L \times D}$ are directly derived from the input data $\mathbf{x} \in \mathbb{R}^{B \times L \times D}$, allowing the model to selectively propagate or forget information based on the current context. Furthermore, Mamba also introduces a hardware-aware Selective Scan algorithm. This algorithm leverages the parallel associative scan property to compute the recurrent state updates efficiently on GPUs, significantly boosting inference speed.

[1] For simplicity, we assume all scales are multiples of the finest one. The framework can be extended to other cases by selecting an appropriate common reference resolution.

[2] We assume square cells for simplicity.

[3] If 0 has a specific meaning, an alternative value will be used.

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

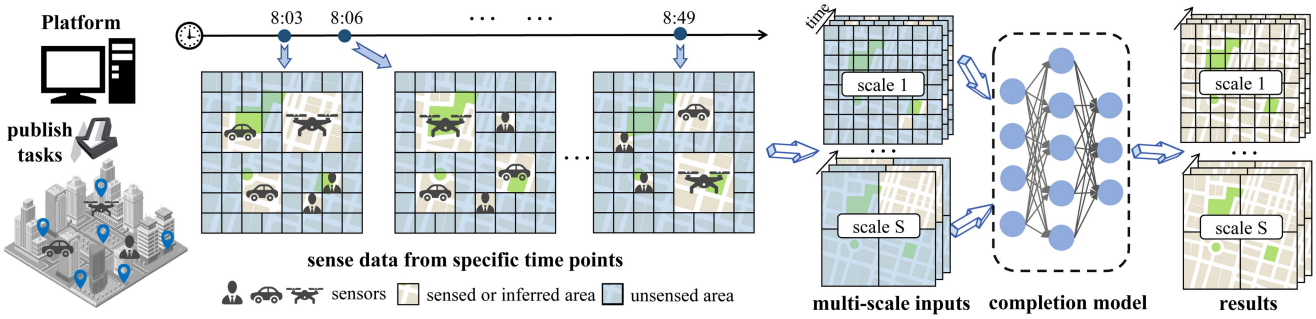


Fig. 2. The workflow of our work: the platform recruits users to collect partial multi-scale data at specific timestamps for data inference.

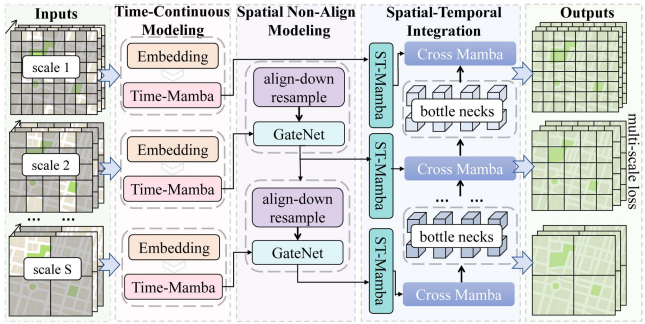


Fig. 3. The structure of the ST-NAI.

IV. Method

A. Overall Structure

As illustrated in Fig. 3, ST-NAI consists of four components: Time-Continuous Modeling, Spatial Non-Align Modeling, Spatial-Temporal Integration, and Output Section. Initially, we conduct preliminary data embedding across all scales to transform raw inputs into latent representations. In the Time-Continuous Modeling phase, we leverage Time-Mamba to capture temporal dynamics based on physical time intervals. Subsequently, the Spatial Non-Align Modeling component utilizes scale-aligned resampling to align features from finer to coarser scales, which are then merged via GateNet to address spatial non-alignment. Since coarse-scale context has already been progressively injected into finer-scale representations through the preceding multi-scale pipeline, Cross-Mamba adopts a coarse-to-fine interaction scheme to avoid redundant reverse fusion and to focus on the finest-scale inference target. Following this, in the Spatial-Temporal Integration phase, ST-Mamba is employed to extract deep spatiotemporal patterns from these integrated features, and Cross-Mamba facilitates global interaction among different scales. Finally, we use the output layer to generate the inference results for the target moment, and the network is trained based on the multi-scale loss.

B. Data Embedding

Data embedding is crucial for the model's performance and accuracy, especially when handling multi-scale spatiotemporal data. Inspired by ST-Transformer [9], our embedding module integrates value embedding, position embedding, spatial information embedding (latitude, longitude, POI), and temporal information embedding (timestamp, holiday). Furthermore, to enhance the model's responsiveness to different scales, we design different parameters for inputs of different scales, while inputs at the same scale share the same parameters. This approach improves the model's adaptability to multi-scale inputs, ensuring that it can effectively handle the diverse and dynamic characteristics of spatiotemporal data across various scales.

The embedding of scale s $\mathbf{X}_{emb}^{(s)} \in \mathbb{R}^{L\times H^{(s)}\times W^{(s)}\times d}$ can be expressed as follows:

$$\mathbf{X}_{emb}^{(s)} = \mathbf{X}_{val}^{(s)} + \mathbf{X}_{p}^{(s)} + \mathbf{S}_{f}^{(s)} + \mathbf{T}_{f}^{(s)}, \tag{9}$$

where $\mathbf{X}_{val}^{(s)}$ represents the value embedding, $\mathbf{X}_{p}^{(s)}$ represents the spatiotemporal position embedding, $\mathbf{S}_{f}^{(s)}$ represents the spatial information embedding, and $\mathbf{T}_{f}^{(s)}$ represents the temporal information embedding.

C. Time-Continuous Modeling

Following data embedding, we obtain latent representations $\mathbf{X}_{emb}^{(s)}$ for each scale s. However, treating these representations simply as an ordered sequence neglects the irregular physical time intervals between observations, leading to the problem of temporal non-alignment. Standard sequence models implicitly assume a fixed interval between time steps t and $t-1$, which fails to capture the varying dynamics of real-world sparse sensing. To address this, we propose the Time-Mamba, which explicitly incorporates physical time intervals into the time-continuous modeling of the Mamba.

To bridge the gap between observation times $\mathcal{T}_{obs}$ and target times $\mathcal{T}_{target}$, we first construct a unified temporal sequence by taking the union of both timestamp sets, denoted as:

$$\mathcal{T}_{seq} = \text{sort}(\mathcal{T}_{obs} \cup \mathcal{T}_{target}), \tag{10}$$

where $\mathcal{T}_{seq}$ represents the union of timestamp sets and sort() represents the chronological ordering. Based on this ordered sequence, we calculate the physical time interval $\mathcal{D}$ as follows:

$$\delta_1 = 0,\ \delta_k = \tau_k - \tau_{k-1},\ k = 2, 3, \ldots, N, \tag{11}$$

$$\mathcal{D} = \{\delta_1,\ \delta_2,\ \cdots,\ \delta_N\}, \tag{12}$$

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

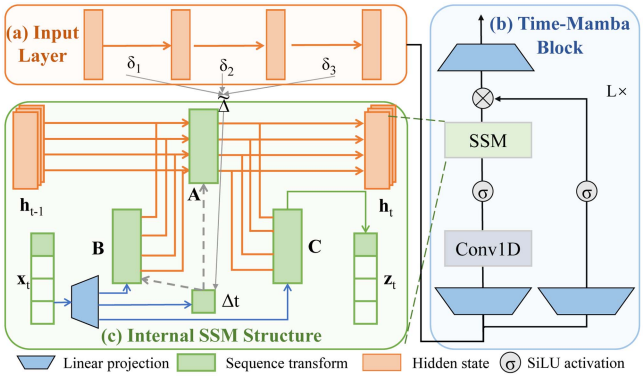


Fig. 4. The structure of the Time-Mamba. Irregular observations are first processed by (a) the Input Layer, then propagated to (b) the Time-Mamba Block for temporal modeling, whose internal mechanism is illustrated in (c) as the Internal SSM Structure.

where N represents the length of the $\mathcal{T}_{seq}$. This ensures that the model evolves continuously through both observation points and target points. For timestamps corresponding to observations ($\tau_k \in \mathcal{T}_{obs}$), the input is the sensed embedding derived in the previous step; for target times without observations ($\tau_k \in \mathcal{T}_{target} \setminus \mathcal{T}_{obs}$), we utilize learnable placeholder vectors to drive the state evolution, allowing the model to infer the state at the target moment based on historical dynamics.

To effectively capture the continuous evolution driven by these irregular intervals, we propose the Time-Mamba architecture. In the standard Mamba, the timescale parameter $\Delta \in \mathbb{R}^{N\times d}$ is derived solely from the input content, making it insensitive to the actual physical intervals $\mathcal{D} \in \mathbb{R}^N$. As shown in Fig. 4, our core insight is to fuse the physical time interval into this parameter to modulate the state transition. Since the distribution of scalar physical intervals differs significantly from high-dimensional content features, direct fusion often leads to training instability. Therefore, we employ a normalization-and-addition strategy. We first project the physical intervals $\mathcal{D} \in \mathbb{R}^N$ into a high-dimensional vector $\mathbf{\Delta}_p \in \mathbb{R}^{N\times d}$ and then apply Layer Normalization to align the statistical distributions of both representations before performing an additive fusion, detailed as follows:

$$\mathbf{\Delta}_p = \text{Linear}(\mathcal{D}), \tag{13}$$

$$\mathbf{\Delta}_f = \text{Softplus}(\text{LayerNorm}(\Delta) + \text{LayerNorm}(\Delta_p)). \tag{14}$$

Here, the physical time information acts as an adaptive bias, adjusting the system's evolution speed based on the actual time intervals, while the Softplus activation ensures the timescale remains positive.

Subsequently, we integrate this fused timescale into the discretization process of the SSM. The discrete state transition matrices $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$, which govern the evolution of the latent state, are updated as follows:

$$\overline{\mathbf{A}} = \exp(\mathbf{\Delta}_f \mathbf{A}), \tag{15}$$

$$\overline{\mathbf{B}} = (\mathbf{\Delta}_f \mathbf{A})^{-1}(\exp(\mathbf{\Delta}_f \mathbf{A}) - \mathbf{I}) \cdot \mathbf{\Delta}_f \mathbf{B}. \tag{16}$$

$$h_t = \bar{\mathbf{A}} h_{t-1} + \bar{\mathbf{B}} x_t, y_t = \mathbf{C} h_t, \tag{17}$$

By substituting these time-aware parameters into the SSM, the Time-Mamba evolves the latent states along the unified timeline $\mathcal{T}_{seq}$. We apply the proposed Time-Mamba to independently learn the temporal dynamics for each scale. Specifically, the latent representation of each scale s is fed into a scale-specific Time-Mamba module, evolving the states along the unified timeline $\mathcal{T}_{seq}$ to capture scale-specific temporal patterns. We denote this full evolved sequence at scale s as $\mathbf{H}^{(s)}_{time} \in \mathbb{R}^{N\times H^{(s)}\times W^{(s)}\times d}$.

Finally, to obtain the representations aligned with the target times, we perform a target state extraction operation. Let $\mathcal{K}_{target} = \{k \mid \tau_k \in \mathcal{T}_{target}\}$ be the set of indices in the unified sequence corresponding to the target times. The final temporally aligned features $\mathbf{X}^{(s)}_{time}$ are extracted from $\mathbf{H}^{(s)}_{time}$, detailed as follows:

$$\mathbf{X}^{(s)}_{time} = \left\{ \mathbf{H}^{(s)}_{time}[k] \mid k \in \mathcal{K}_{target} \right\}, \tag{18}$$

where $\mathbf{X}^{(s)}_{time} \in \mathbb{R}^{T\times H^{(s)}\times W^{(s)}\times d}$ denotes the temporally aligned representations. Through the Time-Mamba mechanism, we effectively bridge the temporal discrepancy between the irregular observation timestamps and the required output moments. This process ensures that the extracted features $\mathbf{X}^{(s)}_{time}$ are strictly aligned with the target times, providing a robust and time-synchronized foundation for the subsequent spatial non-aligned modeling.

D. Spatial Non-Align Modeling

Following the time-continuous modeling, we obtain the temporally aligned representations $\mathbf{X}^{(s)}_{time}$ for each scale. The subsequent crucial step is to enable effective interaction of representations across different scales. However, direct representation fusion is hindered by the mismatched spatial dimensions resulting from the diversity of sensing devices. A general approach to address this is to employ standard convolutional layers to adjust the representation maps to the same dimension before interaction. Nevertheless, in real-world spatially non-aligned scenarios, this standard approach is often inadequate.

For instance, consider downsampling a 5×5 data representation to a 3×3 size. While a convolution with a stride of 2, kernel size of 2, and padding of 1 can technically produce a 3×3 output, it fails to respect the geometric reality that each output cell should correspond to an area of approximately $5/3 \times 5/3$ in the original representation. Standard convolutions, limited by integer operations, cannot achieve this non-integer mapping, leading to spatial misalignments where the representations are structurally distorted. To fundamentally resolve this, we propose a two-step mechanism: first, we employ Scale-Adjusted Resampling to ensure precise geometric alignment; second, we utilize GateNet to effectively fuse the aligned features with the current scale representations.

1) Scale-Adjusted Resampling: To achieve precise alignment between non-integer scales, we utilize the Least Common Multiple of the spatial dimensions. As illustrated in Fig. 5, when downsampling from a fine scale (e.g., 3×3) to a coarse scale (e.g., 2×2), instead of forcing a direct convolution, we first

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

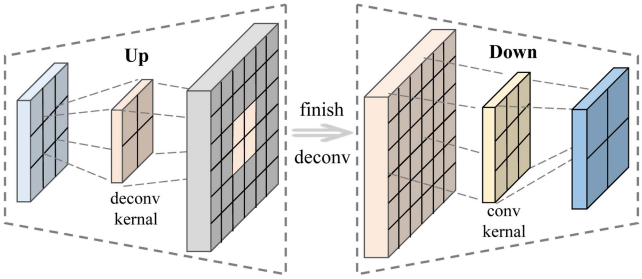


Fig. 5. The example of the Scale-Adjusted Resampling.

upsample the source data to their LCM size (6×6) using a deconvolution operation. This ensures that the spatial grid is perfectly divisible by the target dimension. Subsequently, we apply a standard convolution to downsample from the LCM size to the target size.

Formally, we construct the spatially aligned feature $\mathbf{X}_{space}^{(s)} \in \mathbb{R}^{T\times H^{(s)}\times W^{(s)}\times d}$ for the current coarse scale s as:

$$H_{lcm}^{(s)}=\text{LCM}\left(H^{(s)},H^{(s-1)}\right),W_{lcm}^{(s)}=\text{LCM}\left(W^{(s)},W^{(s-1)}\right), \tag{19}$$

$$\mathbf{H}_{space}^{(s)} = \text{Deconv}\left(\mathbf{X}_{time}^{(s-1)}, \text{kernel}=\left(\frac{H_{lcm}^{(s)}}{H^{(s-1)}}, \frac{W_{lcm}^{(s)}}{W^{(s-1)}}\right)\right), \tag{20}$$

$$\mathbf{X}_{space}^{(s)} = \text{Conv}\left(\mathbf{H}_{space}^{(s)}, \text{ kernel} = \left(\frac{H_{lcm}^{(s)}}{H^{(s)}}, \frac{W_{lcm}^{(s)}}{W^{(s)}}\right)\right), \tag{21}$$

where LCM() denotes the Least Common Multiple function. This operation ensures that the fine-grained information is geometrically aligned with the coarse-grained grid before fusion, effectively addressing the challenges of spatial non-alignment.

2) GateNet Fusion: Having geometrically aligned the representations from the finer scale, the next objective is to merge them with the original representations $\mathbf{X}_{time}^{(s)}$ of the current scale. This interaction is motivated by the complementary nature of the data: $\mathbf{X}_{space}^{(s)}$ provides high-resolution spatial details, while $\mathbf{X}_{time}^{(s)}$ preserves the native temporal dynamics and global context of the current scale. However, since these two sets of representations originate from distinct distributions—one processed through resampling layers and the other derived directly from sensing—simple addition or concatenation may lead to semantic conflict or information redundancy.

To address this, we introduce the GateNet mechanism to dynamically control the information flow. Instead of treating both inputs equally, GateNet learns a context-aware soft mask to weigh the contribution of the aligned spatial details against the original temporal features at a pixel-wise level. Specifically, we concatenate the aligned representations and the current scale representations to generate a gate map $\mathbf{G}_{align}^{(s)} \in \mathbb{R}^{T\times H^{(s)}\times W^{(s)}\times d}$ via a linear projection:

$$\mathbf{G}_{align}^{(s)} = \sigma\left(\text{Linear}\left(\text{Concat}\left(\mathbf{X}_{space}^{(s)}, \mathbf{X}_{time}^{(s)}\right)\right)\right), \tag{22}$$

where σ denotes the sigmoid function, ensuring the gate values fall within $(0, 1)$. The final spatially integrated feature $\mathbf{X}_{align}^{(s)} \in \mathbb{R}^{T\times H^{(s)}\times W^{(s)}\times d}$ is then obtained by:

$$\mathbf{X}_{align}^{(s)} = \mathbf{G}_{space}^{(s)} \odot \mathbf{X}_{space}^{(s)} + (1 - \mathbf{G}_{space}^{(s)}) \odot \mathbf{X}_{time}^{(s)}. \tag{23}$$

Through this selective interaction mechanism, the model adaptively absorbs beneficial spatial details from the finer scale while retaining critical temporal information from the current scale, establishing a robust foundation for the subsequent deep spatiotemporal pattern extraction.

E. Spatial-Temporal Integration

After addressing the non-alignment issues in both temporal and spatial dimensions, we obtain the aligned representations $\mathbf{X}_{align}^{(s)}$ for each scale. However, these features are primarily processed within their respective domains. To achieve comprehensive data inference, it is crucial to deeply integrate spatiotemporal information within each scale and facilitate global interaction across scales. To this end, we first propose the ST-Mamba to integrate spatiotemporal information and extract deep spatiotemporal correlations within each scale. Subsequently, we utilize Cross-Mamba to realize inter-scale spatiotemporal information interaction, ensuring the comprehensive aggregation of global contexts across the entire multi-scale system.

1) St-Mamba: Currently, attention-based architectures, such as Transformers, are commonly employed to capture spatiotemporal relationships due to their comprehensive modeling capabilities. However, the quadratic complexity of Transformers leads to substantial computational overhead. While some attempts have been made to deploy lighter variants, these often compromise performance. Consequently, Mamba has emerged as a promising alternative, offering linear computational efficiency.

Nevertheless, Mamba is inherently designed for sequential data, where the processing order is critical; later nodes in the sequence naturally incorporate the receptive fields of preceding nodes. While this causal property aligns perfectly with temporal data which follows a chronological evolution, it poses a significant challenge for spatial data. In the spatial domain, nodes are topologically connected without a natural sequential order. Imposing a single scanning direction introduces an artificial inductive bias, preventing the model from capturing omnidirectional spatial dependencies effectively.

To address this limitation and make the Mamba sensitive to spatial information, we propose the ST-Mamba architecture. As shown in Fig. 6, this method first employs a Space-Sensitive Block to acquire global spatial perception. Subsequently, we design a Multi-Directional Spatial-Adjacent Scan mechanism followed by a Hierarchical Fusion strategy. This design enables the model to comprehensively learn spatial dependencies from multiple orientations and adaptively integrate them based on the data characteristics.

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

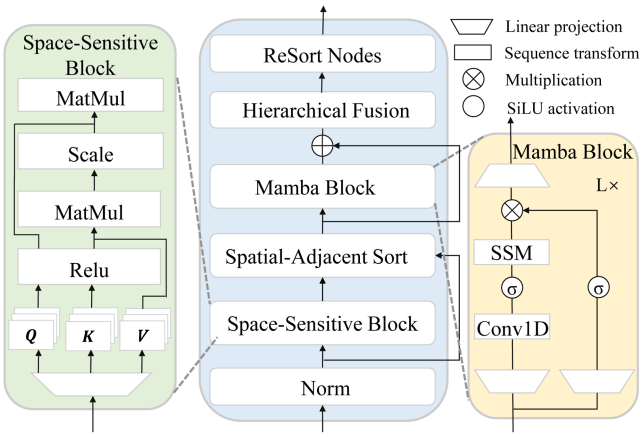


Fig. 6. The structure of the ST-Mamba.

a) Space-Sensitive Block: In spatiotemporal data, the importance of spatial nodes varies dynamically based on their functional roles and real-time states. Simply feeding flattened nodes into Mamba treats them equally in the initial state, ignoring their global semantic relevance. To overcome this, we first employ a Space-Sensitive Block to assign a global receptive field to each node. We utilize a linear spatial-attention mechanism to calculate the relevance scores among all spatial nodes within the same time slice. This ensures that each node is aware of the global spatial context before sequential processing. The process is expressed as:

$$\mathbf{Q}_{align}^{(s)}=\text{Relu}\left(\mathbf{X}_{align}^{(s)}\mathbf{W}_{Q}^{(s)}\right),\mathbf{K}_{align}^{(s)}=\text{Relu}\left(\mathbf{X}_{align}^{(s)}\mathbf{W}_{K}^{(s)}\right), \tag{24}$$

$$\mathbf{V}_{align}^{(s)}=\mathbf{X}_{align}^{(s)}\mathbf{W}_{V}^{(s)}, \tag{25}$$

$$\mathbf{S}_{align}^{(s)}=\mathbf{K}_{align}^{(s)\top}\mathbf{V}_{align}^{(s)},\mathbf{H}_{st}^{(s)}=\mathbf{Q}_{align}^{(s)}\mathbf{S}_{align}^{(s)}, \tag{26}$$

where $\mathbf{W}_{Q}^{(s)},\mathbf{W}_{K}^{(s)},\mathbf{W}_{V}^{(s)}\in\mathbb{R}^{d\times d}$ are learnable projection matrices. Note that the attention is computed within each time slice over spatial tokens. Meanwhile, this formulation follows the standard linear-attention computation, which avoids explicitly constructing the attention matrix $\mathbf{Q}_{align}^{(s)}\mathbf{K}_{align}^{(s)\top}$ and reduces the spatial interaction cost to linear complexity.

b) Multi-Directional Spatially-Adjacent Scan: Having established a global context, the next step is to capture fine-grained local dependencies. As discussed, a single scanning order introduces directional bias. To capture spatial correlations without such bias, we propose a Multi-Directional Spatially-Adjacent Scan strategy. As illustrated in Fig. 7, we unfold the 2D spatial grid into 1D sequences along four distinct directions: horizontal forward ($\rightarrow$), horizontal backward ($\leftarrow$), vertical downward ($\downarrow$), and vertical upward ($\uparrow$). This ensures that the Mamba model can capture dependencies from multiple spatial perspectives while strictly maintaining spatial adjacency.

Let $\mathcal{F}_{dir}$ denote the function that flattens the 2D grid into a sequence according to the direction $dir\in\{\rightarrow,\leftarrow,\downarrow,\uparrow\}$. The

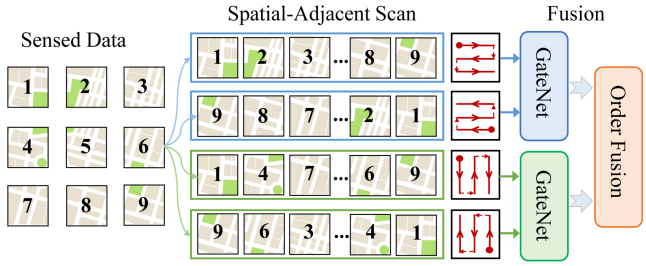


Fig. 7. The structure of the Spatial-Adjacent Scan.

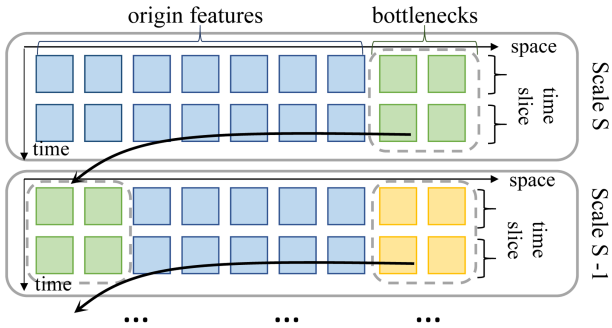


Fig. 8. The example of the Cross-Mamba.

globally aware features $\mathbf{H}_{st}^{(s)}$ are processed as:

$$\mathbf{H}_{dir}^{(s)}=\text{Mamba}(\mathcal{F}_{dir}(\mathbf{H}_{st}^{(s)})),\quad dir\in\{\rightarrow,\leftarrow,\downarrow,\uparrow\}. \tag{27}$$

Here, four independent Mamba blocks are employed to process these sequences in parallel, extracting directional patterns $\mathbf{H}_{dir}^{(s)}$ from each respective viewpoint.

c) Hierarchical Fusion: Finally, we integrate the features from different scanning directions to form a comprehensive spatial representation. To avoid information redundancy and semantic conflict between opposing directions, we employ a Hierarchical Fusion strategy using the previously defined GateNet. We first fuse opposite directions pair-wisely to form bi-directional representations, detailed as follows:

$$\mathbf{G}_{hor}^{(s)}=\sigma(\text{Linear}(\text{Concat}(\mathbf{H}_{\rightarrow}^{(s)},\mathbf{H}_{\leftarrow}^{(s)}))), \tag{28}$$

$$\mathbf{H}_{hor}^{(s)}=\mathbf{G}_{hor}^{(s)}\odot\mathbf{H}_{\rightarrow}^{(s)}+(1-\mathbf{G}_{hor}^{(s)})\odot\mathbf{H}_{\leftarrow}^{(s)}, \tag{29}$$

$$\mathbf{G}_{ver}^{(s)}=\sigma(\text{Linear}(\text{Concat}(\mathbf{H}_{\downarrow}^{(s)},\mathbf{H}_{\uparrow}^{(s)}))), \tag{30}$$

$$\mathbf{H}_{ver}^{(s)}=\mathbf{G}_{ver}^{(s)}\odot\mathbf{H}_{\downarrow}^{(s)}+(1-\mathbf{G}_{ver}^{(s)})\odot\mathbf{H}_{\uparrow}^{(s)}. \tag{31}$$

Then, we fuse the resulting horizontal and vertical features:

$$\mathbf{G}_{st}^{(s)}=\sigma(\text{Linear}(\text{Concat}(\mathbf{H}_{hor}^{(s)},\mathbf{H}_{ver}^{(s)}))), \tag{32}$$

$$\mathbf{X}_{st}^{(s)}=\mathbf{G}_{st}^{(s)}\odot\mathbf{H}_{hor}^{(s)}+(1-\mathbf{G}_{st}^{(s)})\odot\mathbf{H}_{ver}^{(s)}. \tag{33}$$

This hierarchical approach effectively synthesizes multi-directional information, dynamically balancing the contribution of each direction. The resulting $\mathbf{X}_{st}^{(s)}\in\mathbb{R}^{T\times H^{(s)}\times W^{(s)}\times d}$ represents a robust, deep spatial embedding that encapsulates both global context and omnidirectional local dependencies.

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: The algorithm of the ST-Mamba.

Input: Data $\mathbf{X}_{align}^{(s)}$ at scale s

Space-Sensitive Block:
$\mathbf{Q}_{align}^{(s)}, \mathbf{K}_{align}^{(s)}, \mathbf{V}_{align}^{(s)} \leftarrow \text{Eq.}(24,25)(\mathbf{X}_{align}^{(s)})$;
$\mathbf{H}_{st}^{(s)} \leftarrow \text{Eq.}(26)(\mathbf{Q}_{align}^{(s)}, \mathbf{K}_{align}^{(s)}, \mathbf{V}_{align}^{(s)})$;
Multi-Directional Spatially-Adjacent Scan:
for *each dir in* $\{\rightarrow, \leftarrow, \downarrow, \uparrow\}$ **do**
  $\mathbf{H}_{dir}^{(s)} \leftarrow \text{Eq.}(27)(\mathbf{H}_{st}^{(s)})$;
Hierarchical Fusion:
$\mathbf{H}_{hor}^{(s)} \leftarrow \text{Eq.}(28,29)(\mathbf{H}_{\rightarrow}^{(s)}, \mathbf{H}_{\leftarrow}^{(s)})$;
$\mathbf{H}_{ver}^{(s)} \leftarrow \text{Eq.}(30,31)(\mathbf{H}_{\downarrow}^{(s)}, \mathbf{H}_{\uparrow}^{(s)})$;
$\mathbf{X}_{st}^{(s)} \leftarrow \text{Eq.}(32,33)(\mathbf{H}_{hor}^{(s)}, \mathbf{H}_{ver}^{(s)})$;
return $\hat{\mathbf{X}}_{st}^{(s)}$.

2) Cross-Mamba: After extracting deep spatiotemporal patterns within each scale via ST-Mamba, capturing correlations between multiple scales is the final and critical step for holistic data inference. A straightforward approach to inter-scale interaction would be to concatenate the feature sequences from all scales into a single long sequence and feed it into a Mamba model. However, this naive strategy faces significant challenges. Mamba is inherently a causal model where information flows from past to future tokens. If we simply concatenate scales (e.g., Scale 1 followed by Scale 2), the earlier scales in the sequence (Scale 1) cannot access information from the later scales (Scale 2) during a single forward pass. This creates a barrier to global interaction, preventing the model from fully leveraging the complementary information distributed across scales.

To overcome these limitations, we draw inspiration from the Bottleneck architecture [29]. The core idea is to restrict the interaction to a small set of shared latent units (bottlenecks) rather than allowing dense pairwise interactions between all nodes. These bottlenecks compress information from one source and distribute it to another. Based on this, we propose Cross-Mamba. Instead of concatenating all data, we maintain separate processing streams for each scale but link them via a shared bottleneck memory. Complementing our spatial modeling strategy, we orchestrate the information flow from the coarsest scale to the finest scale. Since coarse-scale context has already been progressively injected into finer-scale representations through the preceding multi-scale pipeline, adopting an additional reverse interaction would introduce redundant fusion while bringing limited benefit to the finest-scale target. This design allows the model to effectively transfer global context across scales while preserving the focus on fine-grained inference.

As illustrated in Fig. 8, since the later nodes in the Mamba architecture can learn from the preceding nodes, we append additional learnable nodes (bottlenecks) $\mathbf{B}^{(s)}$ to the end of the original sequence $\mathbf{X}_{st}^{(s)}$. As the ST-Mamba processes the sequence, these tail tokens compress the global semantic information into a compact memory $\mathbf{E}^{(s)}$. Subsequently, we prepend the learned content to finer-scale nodes, allowing them to learn the receptive fields of coarser scales. Similarly, the final empty nodes obtain global receptive fields, and this process is repeated down to the finest scale. We place bottleneck nodes at the end of each time dimension to capture all spatial information within each time slice. This process is formulated as follows:

$$\left[\mathbf{X}_{\text{out}}^{(S)}, \mathbf{E}^{(S)}\right] = \text{STM}\left([\mathbf{X}_{st}^{(S)}, \mathbf{B}^{(S)}]\right), \tag{34}$$

$$\left[_, \mathbf{X}_{\text{out}}^{(s)}, \mathbf{E}^{(s)}\right] = \text{STM}\left([\mathbf{E}^{(s+1)}, \mathbf{X}_{st}^{(s)}, \mathbf{B}^{(s)}]\right), \tag{35}$$

$$\left[_, \mathbf{X}_{\text{out}}^{(1)}\right] = \text{STM}\left([\mathbf{E}^{(2)}, \mathbf{X}_{st}^{(1)}]\right), \tag{36}$$

where _ represents an unused output tensor that is discarded after the STM operation and not involved in subsequent computations, STM represents the ST-Mamba method, $\mathbf{B}^{(s)} \in \mathbb{R}^{T \times n_b \times d}$ represents bottleneck nodes of scale s, $\mathbf{E}^{(s)} \in \mathbb{R}^{T \times n_b \times d}$ represents the learned information, n_b represents the number of bottleneck tokens appended at each time step, and $\mathrm{s} \in [2, \mathrm{S}-1]$. Through this mechanism, Cross-Mamba achieves efficient global interaction with linear complexity. It ensures that the coarse-scale representations are not isolated but are enriched with summarized details from finer scales, creating a coherent and hierarchically integrated multi-scale system.

F. Multi-Scale Loss

For each scale, we use a linear layer to obtain the final inference results and consider the inference effect of each scale during the final loss calculation. Formally:

$$\hat{\mathbf{Z}}^{(s)} = \text{Linear}(\mathbf{X}_{\text{out}}^{(s)}), \tag{37}$$

$$\mathcal{L} = \sum\nolimits_{s=1}^{S} C_s \times \text{MSE}(\hat{\mathbf{Z}}^{(s)}, \mathbf{Z}^{(s)}), \tag{38}$$

where C_s is a constant denoting the weight of the s-th scale result. We show the ST-NAI algorithm in Algorithm 2.

V. Theoretical Analysis

In this section, we provide a theoretical analysis of the proposed framework. We prove the necessity of temporal consistency of Time-Mamba in Theorem 1, the multi-scale data utilization in Theorem 2, and the computational efficiency in Theorem 3.

Theorem 1: Under irregularly sampled systems, Time-Mamba ensures consistent state evolution by explicitly approximating the analytical solution of the continuous differential equation governing the latent dynamics.

Proof: Consider the latent state $h(t)$ governed by a linear continuous differential equation:

$$\frac{dh(t)}{dt} = \mathbf{A}h(t) + \mathbf{B}x(t). \tag{39}$$

To determine the state evolution between two irregular observation timestamps t_{k-1} and t_k (with physical interval $\delta_k = t_k - t_{k-1}$), multiplying both sides of (39) by the matrix exponential $e^{-\mathbf{A}t}$ yields:

$$e^{-\mathbf{A}t}\frac{dh(t)}{dt} - e^{-\mathbf{A}t}\mathbf{A}h(t) = e^{-\mathbf{A}t}\mathbf{B}x(t). \tag{40}$$

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2: The algorithm of the ST-NAI.

Input: Sparse observation $\mathbf{X}$ with timestamps $\mathcal{T}_{obs}$, target times $\mathcal{T}_{target}$, max epochs MAX_ITER

while *not convergent* ***and*** $count < MAX_ITER$ **do**
  Data Embedding:
  $\mathbf{X}_{emb}^{(s)} \leftarrow \text{Eq.}(9)(\mathbf{X}^{(s)})$;
  Time-Continuous Modeling:
  $\mathcal{T}_{seq} \leftarrow \text{Eq.}(10)(\mathcal{T}_{obs}, \mathcal{T}_{target})$;
  $\mathcal{D} \leftarrow \text{Eq.}(11, 12)(\mathcal{T}_{seq})$;
  $\mathbf{H}_{time}^{(s)} \leftarrow \text{Eq.}(13-17)(\mathcal{D}, \mathbf{X}_{emb}^{(s)})$;
  $\mathbf{X}_{time}^{(s)} \leftarrow \text{Eq.}(18)(\mathbf{H}_{time}^{(s)})$;
  Spatial Non-Align Modeling:
  $\mathbf{X}_{align}^{(0)} \leftarrow \mathbf{X}_{time}^{(0)}$;
  for *each* s *in* $[2, S]$ **do**
    $\mathbf{X}_{space}^{(s)} \leftarrow \text{Eq.}(19-21)(\mathbf{X}_{time}^{(s)})$
    $\mathbf{X}_{align}^{(s)} \leftarrow \text{Eq.}(22, 23)(\mathbf{X}_{time}^{(s)}, \mathbf{X}_{space}^{(s)})$;
  Spatial-Temporal Integration:
  for *each* s *in* $[1, S]$ **do**
    $\mathbf{X}_{st}^{(s)} \leftarrow \text{ST-Mamba}(\mathbf{X}_{align}^{(s)})$;
  $[\mathbf{X}_{out}^{(S)}, \mathbf{E}^{(S)}] \leftarrow \text{Eq.}(34)([\mathbf{X}_{st}^{(S)}, \mathbf{0}])$;
  for *each* s *in* $[S-1, 2]$ **do**
    $[\mathbf{X}_{out}^{(s)}, \mathbf{E}^{(s)}] \leftarrow \text{Eq.}(35)([\mathbf{E}^{(s+1)}, \mathbf{X}_{st}^{(s)}, \mathbf{0}])$;
  $\mathbf{X}_{out}^{(1)} \leftarrow \text{Eq.}(36)([\mathbf{E}^{(2)}, \mathbf{X}_{st}^{(1)}])$;
  Output:
  for *each* s *in* $[1, S]$ **do**
    $\hat{\mathbf{Z}}^{(s)} \leftarrow \text{Eq.}(37)(\mathbf{X}_{out}^{(s)})$;
  $\mathcal{L} \leftarrow \text{Eq.}(38)(\hat{\mathbf{Z}}, \mathbf{Z})$;
return $\hat{\mathbf{Z}}^{(1)}$.

Recognizing the left side as the derivative of $\frac{d}{dt}(e^{-\mathbf{A}t}h(t))$, we integrate both sides over the interval $[t_{k-1}, t_k]$:

$$\int_{t_{k-1}}^{t_k} \frac{d}{d\tau}\left(e^{-\mathbf{A}\tau}h(\tau)\right)d\tau = \int_{t_{k-1}}^{t_k} e^{-\mathbf{A}\tau}\mathbf{B}x(\tau)d\tau. \tag{41}$$

Applying the Fundamental Theorem of Calculus, we obtain:

$$e^{-\mathbf{A}t_k}h(t_k) - e^{-\mathbf{A}t_{k-1}}h(t_{k-1}) = \int_{t_{k-1}}^{t_k} e^{-\mathbf{A}\tau}\mathbf{B}x(\tau)d\tau. \tag{42}$$

Multiplying both sides by $e^{\mathbf{A}t_k}$ and utilizing the property $e^{\mathbf{A}t_k}e^{-\mathbf{A}t_{k-1}} = e^{\mathbf{A}(t_k - t_{k-1})} = e^{\mathbf{A}\delta_k}$, we arrive at the exact analytical solution:

$$h(t_k) = \underbrace{e^{\mathbf{A}\delta_k}}_{\overline{\mathbf{A}}}h(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{\mathbf{A}(t_k-\tau)}\mathbf{B}x(\tau)d\tau. \tag{43}$$

This derivation rigorously proves that under the Zero-Order Hold (ZOH) assumption, the discrete state transition matrix $\overline{\mathbf{A}}$ must be an exponential function scaled by the physical time interval δ_k.

Although Mamba extends the traditional LTI system to the time-varying system by making parameters context-aware (i.e., $\mathbf{A} \to \mathbf{A}(x_k)$ or $\Delta \to \Delta(x_k)$), it does not alter the underlying physics of discretization. The state transition magnitude must still scale with the elapsed physical time.

Time-Mamba constructs its discretization timescale Δ_f by fusing the content-driven features with the physical interval δ_k. The resulting discretized state transition matrix in Time-Mamba is formulated as follows:

$$\overline{\mathbf{A}} = \exp(\Delta_f(\delta_k, x_k) \cdot \mathbf{A}). \tag{44}$$

Because Δ_f explicitly takes δ_k as an input variable, the neural network has the structural capacity to learn a mapping $\Delta_f \approx \alpha \cdot \delta_k$ (where α is a scaling factor). Consequently, the modeled transition $\overline{\mathbf{A}}$ can asymptotically approximate the theoretical ground truth $\exp(\delta_k\mathbf{A})$.

In contrast, models that do not incorporate δ_k into the exponent (such as standard sequence models) inherently lack this structural capacity, leading to an approximation error that cannot be eliminated by parameter optimization alone. Thus, Time-Mamba theoretically ensures consistency with the continuous dynamics. □

Definition 1: Let $\mathbf{X}_{multi}$ denote the multi-scale data representation information, and let $\mathbf{X}_{single}$ denote the input data information at only single scale.

Theorem 2: The mutual information between $\mathbf{X}_{multi}$ and the target variable $\mathbf{Z}$ consistently exceeds the mutual information between $\mathbf{X}_{single}$ and $\mathbf{Z}$.

Proof: Mutual information is defined as follows:

$$\begin{aligned} I(\mathbf{Z}; \mathbf{X}_{multi}) &= H(\mathbf{Z}) - H(\mathbf{Z} \mid \mathbf{X}_{multi}), \\ I(\mathbf{Z}; \mathbf{X}_{single}) &= H(\mathbf{Z}) - H(\mathbf{Z} \mid \mathbf{X}_{single}), \end{aligned} \tag{45}$$

where $H(\mathbf{Z})$ denotes the entropy of $\mathbf{Z}$, and $H(\mathbf{Z} \mid \mathbf{X}_{multi})$ and $H(\mathbf{Z} \mid \mathbf{X}_{single})$ represent the conditional entropy given multi-scale and single-scale data, respectively.

Since $\mathbf{X}_{multi}$ is constructed by integrating data from multiple scales, it inherently encapsulates all the information in $\mathbf{X}_{single}$ and additional features from other scales. This additional information reduces the uncertainty of the target variable $\mathbf{Z}$, leading to:

$$H(\mathbf{Z} \mid \mathbf{X}_{multi}) \leq H(\mathbf{Z} \mid \mathbf{X}_{single}). \tag{46}$$

Combining (45) and (46), we obtain:

$$I(\mathbf{Z}; \mathbf{X}_{multi}) \geq I(\mathbf{Z}; \mathbf{X}_{single}), \tag{47}$$

indicating that the multi-scale representation improves the mutual information with the target variable $\mathbf{Z}$ compared to single-scale data. This result highlights the importance of utilizing multi-scale data for inference. □

Definition 2: Denote $L^{(s)} = H/g^{(s)} \times W/g^{(s)}$ as the number of spatial nodes at scale s, N as the length of the unified temporal sequence, and d as the model hidden dimension.

Theorem 3: The computational complexity of the proposed ST-Mamba is linear with respect to both the time sequence length N and the total number of spatial nodes $\sum L^{(s)}$.

Proof: In ST-Mamba, the sequential state update inherited from Mamba is inherently linear, since it processes the input as a recurrent state-space computation and scans the sequence once, leading to a cost that scales linearly with the temporal length N.

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

In addition, the Space-Sensitive Block is applied within each time slice over $L^{(s)}$ spatial tokens to inject global spatial context before sequential modeling. We implement this block using a linear attention formulation. Concretely, for a fixed time slice, after projecting the spatial tokens to obtain $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, we first aggregate value features through the keys by computing an intermediate summary of size $d \times d$, which costs $\mathcal{O}(L^{(s)} \cdot d^2)$. We then multiply the queries with this summary to obtain the final outputs, which also costs $\mathcal{O}(L^{(s)} \cdot d^2)$. Importantly, this computation avoids forming pairwise interactions between all token pairs and thus scales linearly with the number of spatial tokens. Therefore, the per-slice complexity of the Space-Sensitive Block is $\mathcal{O}(L^{(s)} \cdot d^2)$ (up to constant projection factors). Since this operation is performed independently across N time steps, the total cost at scale s is $\mathcal{O}(N \cdot L^{(s)} \cdot d^2)$.

Consequently, summing over all S scales, the overall complexity of ST-Mamba is $\mathcal{O}(N \cdot \sum_{s=1}^{S} L^{(s)} \cdot d^2)$. Since the hidden dimension d is a model constant, the computational complexity is linear with respect to both the time sequence length N and the total number of spatial nodes $\sum_{s=1}^{S} L^{(s)}$. □

Theorem 4: When all spatial scales are defined as integer multiples of a common reference finest grid, i.e., each scale factor $g^{(s)}$ is an integer divisor of both H and W, the spatial resampling in the scale-adjusted alignment has linear computational complexity with respect to the spatial grid size.

Proof: In scale-adjusted spatial resampling, the intermediate aligned grid between two scales s and s' is determined by the least common multiple (LCM) of their spatial resolutions:

$$H^{lcm} = \mathrm{lcm}(H^{(s)}, H^{(s')}), W^{lcm} = \mathrm{lcm}(W^{(s)}, W^{(s')}). \quad (48)$$

Since

$$H^{(s)} = H/g^{(s)}, H^{(s')} = H/g^{(s')}, \quad (49)$$

and $g^{(s)}$, $g^{(s')}$ are integer divisors of H, we have:

$$H^{lcm} = \mathrm{lcm}\left(\frac{H}{g^{(s)}}, \frac{H}{g^{(s')}}\right) = \frac{H}{\gcd(g^{(s)}, g^{(s')})} \leq H. \quad (50)$$

Similarly,

$$W^{lcm} = \mathrm{lcm}\left(\frac{W}{g^{(s)}}, \frac{W}{g^{(s')}}\right) = \frac{W}{\gcd(g^{(s)}, g^{(s')})} \leq W. \quad (51)$$

Therefore, the aligned intermediate grid is always bounded by the finest spatial grid:

$$H^{lcm} \times W^{lcm} \leq H \times W. \quad (52)$$

For a fixed set of spatial scales, the local resampling kernel size is determined only by the corresponding scale ratios and thus contributes a constant factor. The spatial resampling operation thus processes at most $H \times W$ spatial locations at each time step, leading to a total complexity of

$$\mathcal{O}(N \cdot H \cdot W \cdot d), \quad (53)$$

which is linear with respect to the spatial grid size.

This assumption is consistent with practical grid-based sensing systems, where different spatial scales are typically defined with respect to a common reference resolution. The reference finest grid does not need to be directly observed and can also be introduced as a virtual grid for unified scale alignment. □

VI. Performance Evaluation

A. Setting

1) Datasets: We conduct experiments on three widely-used benchmark datasets in Sparse MCS, namely Air-Quality, Weather, and Traffic. These datasets represent classical grid-based sensing scenarios in MCS and naturally exhibit multi-scale spatial structures, making them well-suited for evaluating spatiotemporal non-aligned inference.

Air-Quality[4] is derived from CHAP [30], [31] and contains meteorological air-quality data. We select PM2.5 and PM10 as the experimental datasets and select 20×20 stations.

Weather[5] contains meteorological data from 2017 to 2018. We select Humidity and Temperature as the experimental datasets and select 24×12 stations for experiments.

Traffic[6] is derived from TaxiBJ [32], which contains meteorological traffic-flow data in Beijing. We select P1 and P2 as the experimental datasets and select 16×16 stations.

2) Data Preprocessing: Spatially, we set the total number of scales to 4. To demonstrate the adaptability to non-aligned compositions, the scales are set to 1, 2, 4, 5 for Air-Quality and 1, 2, 3, 6 for Weather. Additionally, for the aligned scenario, Traffic scales are set to 1, 2, 4, 8. The downsampling rules follow physical properties (averaging or summation).

Temporally, we simulate stochastic sampling by randomly selecting fine-grained observations, with target times defined as the last moment of each period. To reflect different modeling paradigms, we prepare two input formats: aggregated averages for traditional discrete methods (simulating real-world MCS systems with compromised precision) and precise timestamped sequences for time-continuous methods. Note that to ensure fairness, we also conduct specific experiments where discrete baselines are equipped with time-sensitive modules to adapt to irregular sampling, allowing for a fair comparison under continuous settings.

3) Baselines: To comprehensively evaluate the proposed framework, we consider five categories of baselines: spatial completion, spatial multi-scale completion, discrete temporal interpolation, continuous temporal modeling, and spatiotemporal completion methods.

Spatial Completion Methods: We consider MF [33], a matrix factorization method; DMF [34], a neural network implementation of matrix factorization; and GCN [35], which utilizes a graph convolutional network for data completion.

Spatial Multi-Scale Methods: We include NAMS [36], a multi-scale completion framework that handles non-aligned compositional relationships through least-common-multiple-based alignment; Self-Attn [37], the self-attention variant of a spatiotemporal pyramid-based multi-scale completion method.

[4]https://nnu.geodata.cn/featured_data.html
[5]https://www.kdd.org/kdd2018/kdd-cup
[6]https://gitee.com/arislee/taxi-bj

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

Discrete Temporal Interpolation Methods: We consider Mamba [13], a linear-time sequence modeling method; Informer [38], which employs sparse attention mechanisms; Timesnet [39], a CNN-based architecture for time series modeling; and SAITS [16], a self-attention-based method for multivariate time series imputation.

Continuous Temporal Methods: We evaluate methods designed for irregular sampling, including ODE-LSTM [25], which generalizes LSTMs with time-continuous dynamics; ODE-RSSM [26], which combines ODEs with recurrent state-space models; log-NCDE [40], which models irregular observations via neural controlled differential equations; and DIFFODE [27], a differentiable hidden-state-enhanced Neural ODE framework.

Spatiotemporal Completion Methods: We compare with ST-BGMC [8], a matrix completion method with spatiotemporal constraints; ST-Transi [9], a Transformer-based approach for Sparse MCS; PDFormer [41], which models propagation delays in spatiotemporal data; and ImputeFormer [17], which combines low-rank priors with Transformer architectures.

4) Experimental Settings: In our experiments, the datasets are divided into training, validation, and test sets in a 7:2:1 ratio. We use Mean Squared Error (MSE) and Mean Absolute Error (MAE) as evaluation metrics. All experiments are conducted under PyTorch and accelerated using a single NVIDIA GeForce RTX 3090 GPU. For model configurations, we set $d = 64$ to balance representation capacity and computational efficiency, and the number of layers is set to 1 to maintain a lightweight architecture. The input length L is set to 64 based on validation performance and to ensure consistent settings across datasets. For multi-scale training, we assign larger weights to finer scales to better preserve high-resolution details, thus $C = \{0.4, 0.3, 0.2, 0.1\}$. We use a bottleneck size of 4 for efficient cross-scale information compression. The sensitivity of C and the bottleneck size are further evaluated in the following experiments.

TABLE I
MAIN NOTATIONS

Symbol	Type	Meaning
S	scalar	Spatial scales number.
L, T, N	scalar	Observed timestamps length, target timestamps length, unified time sequence length.
H, W	scalar	Spatial grid height, and spatial grid width.
$g^{(s)}$,$H^{(s)}$,$W^{(s)}$	scalar	Scale factor and spatial grid height/width at scale s.
$\mathcal{T}_{obs}$, $\mathcal{T}_{target}$, $\mathcal{T}_{seq}$	set	Observed timestamps, target timestamps, and unified timestamps.
$\mathbf{X}^{(s)}$, $\mathbf{Y}^{(s)}$	$\mathbb{R}^{L\times H^{(s)}\times W^{(s)}}$	Observed data and ground-truth at observed timestamps (length L) at scale s, respectively.
$\mathbf{M}^{(s)}$	$\{0,1\}^{L\times H^{(s)}\times W^{(s)}}$	Observation mask at scale s.
$\mathbf{Z}^{(s)}$, $\hat{\mathbf{Z}}^{(s)}$	$\mathbb{R}^{T\times H^{(s)}\times W^{(s)}}$	Ground-truth and inferred results of target timestamps at scale s.
$\mathbf{X}^{(s)}_{emb}$	$\mathbb{R}^{L\times H^{(s)}\times W^{(s)}\times d}$	Embedded representation of observed inputs (length L) at scale s.
$\mathbf{X}^{(s)}_{time}$,$\mathbf{X}^{(s)}_{space}$, $\mathbf{X}^{(s)}_{align}$	$\mathbb{R}^{T\times H^{(s)}\times W^{(s)}\times d}$	Temporal-view features, spatial-view features, and fused spatiotemporal features aligned on target timestamps (length T) at scale s.
$\mathbf{X}^{(s)}_{st}$,$\mathbf{X}^{(s)}_{out}$	$\mathbb{R}^{T\times H^{(s)}\times W^{(s)}\times d}$	Spatiotemporal features at scale s after ST-Mamba integration, and output representations at scale s after cross-scale fusion.

B. Inference Performance

1) Performance Under the Same Sensed Quantity: We conduct experiments with a sense ratio of 0.01 for each scale input, where the **sense ratio** denotes the fraction of observed entries among all spatiotemporal grid cells in a scale. Since most baselines are single-scale, we feed them the finest-scale data with the same total sensed quantity as our multi-scale inputs for fair comparison. As shown in Table II, ST-NAI achieves the lowest error across all datasets. The two spatial multi-scale baselines, i.e., NAMS and Self-Attn, improve over purely spatial completion methods by exploiting cross-scale information, but they still cannot effectively handle temporal non-alignment. Discrete methods suffer from temporal information loss due to time-slot aggregation. In contrast, continuous ODE-based methods (i.e., ODE-LSTM and ODE-RSSM) parameterize latent dynamics via neural ordinary differential equations, which may favor smoother trajectories and can be challenged by complex, non-smooth real-world variations on high-dimensional grids, leading to over-smoothed representations. By explicitly modeling both spatial and temporal misalignments, ST-NAI avoids these limitations and achieves consistently better inference performance.

Furthermore, we evaluate robustness by increasing the finest-scale sense ratios from 0.01 to 0.05. As shown in Fig. 9, ST-NAI consistently yields the lowest MSE across all datasets and all sense ratios, and its error decreases steadily as more observations become available. Notably, the superiority is most pronounced in Air-Quality and Weather, where sparse and irregular samples make temporal alignment and cross-scale information particularly important, while on Traffic the improvement remains stable despite higher dynamics.

2) Performance Under Multi-Scale Sense Ratios: Fig. 10 further explores the impact on model inference performance as the sense ratios of two scales are increased. Due to experimental constraints, this part of the study is conducted only on the PM2.5 dataset. The results show that the overall performance of the model tends to increase as the scale increases and that finer scales provide better results for the same sensing range.

3) Time-Continuous Fairness Comparison: To ensure a fair comparison under irregular sampling, we further equip representative discrete baselines with time-continuous handling and re-evaluate their inference performance. As shown in Fig. 11, this treatment consistently improves the competing methods, confirming that ignoring temporal non-alignment can be a major source of error. Nevertheless, ST-NAI remains the best across all datasets and sense ratios, indicating that simple time-aware adaptations are still insufficient to model continuous dynamics in high-dimensional grids. This is because ST-NAI explicitly integrates physical time gaps through Time-Mamba and jointly

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

TABLE II
INFERENCE PERFORMANCE UNDER SIX DATASETS

Data	Air-Quality				Weather				Traffic			
Set	PM2.5		PM10		Humidity		Temperature		P1		P2	
Models	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
MF	1.066	0.808	1.085	0.814	0.877	0.730	0.486	0.515	0.491	0.459	0.507	0.465
DMF	0.986	0.802	0.985	0.802	0.970	0.795	0.948	0.809	0.941	0.729	0.936	0.727
GCN	0.197	0.316	0.173	0.297	0.154	0.303	0.027	0.126	1.188	0.828	0.938	0.739
NAMS	0.220	0.355	0.170	0.291	0.295	0.416	0.030	0.135	0.294	0.367	0.240	0.304
Self-Attn	0.211	0.324	0.183	0.301	0.114	0.249	0.026	0.125	0.262	0.326	0.208	0.297
Mamba	0.221	0.329	0.185	0.305	0.187	0.330	0.050	0.178	0.298	0.360	0.224	0.315
Informer	0.214	0.321	0.182	0.300	0.182	0.324	0.042	0.162	0.298	0.359	0.223	0.313
Timesnet	0.220	0.334	0.183	0.301	0.151	0.299	0.054	0.183	0.266	0.328	0.205	0.293
SAITS	0.213	0.325	0.186	0.302	0.117	0.259	0.028	0.128	0.257	0.318	0.214	0.298
ODE-LSTM	0.862	0.753	0.775	0.721	1.552	0.999	1.237	1.010	1.109	0.862	0.887	0.767
ODE-RSSM	0.861	0.750	0.768	0.716	1.607	1.017	1.250	1.017	1.107	0.862	0.888	0.767
log-NCDE	0.472	0.485	0.600	0.583	1.004	0.809	0.668	0.683	1.055	0.823	0.847	0.732
DIFFODE	0.487	0.500	0.618	0.601	1.035	0.834	0.688	0.704	1.087	0.848	0.873	0.755
ST-BGMC	1.022	0.816	1.027	0.817	1.029	0.813	1.014	0.822	1.020	0.745	1.016	0.744
ST-Transi	0.212	0.324	0.183	0.300	0.168	0.311	0.046	0.174	0.276	0.337	0.225	0.316
PDFormer	0.209	0.327	0.183	0.403	0.159	0.306	0.045	0.171	0.259	0.321	0.208	0.297
ImputeFormer	0.245	0.340	0.213	0.322	0.105	0.259	0.105	0.259	0.292	0.351	0.273	0.352
ST-NAI (Our)	**0.026**	**0.108**	**0.022**	**0.101**	**0.102**	**0.239**	**0.025**	**0.123**	**0.252**	**0.313**	**0.198**	**0.286**

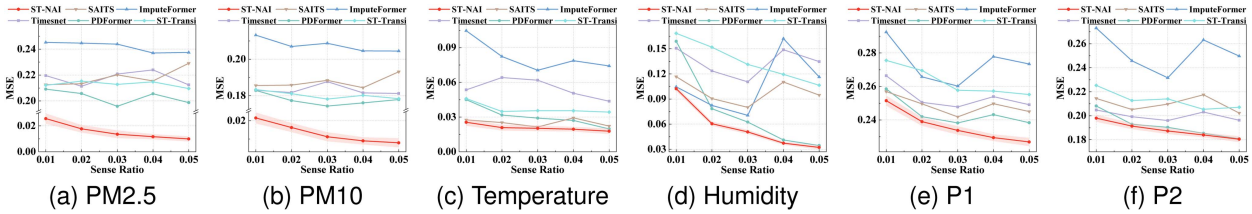


Fig. 9. Inference performance under different sense ratios with multi-scale inputs. (a) PM2.5. (b) PM10. (c) Temperature. (d) Humidity. (e) P1. (f) P2.

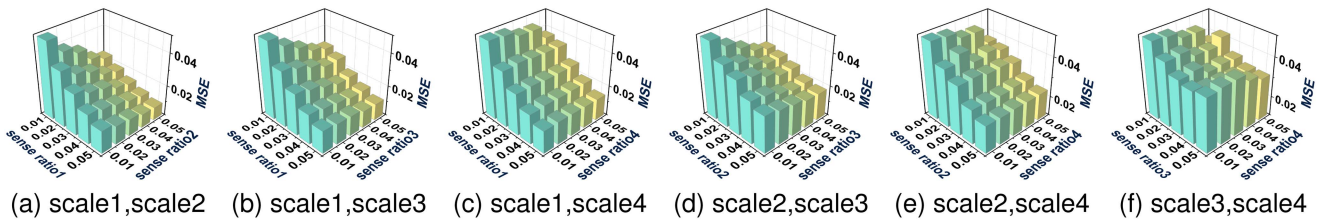


Fig. 10. Inference performance under two different scales. (a) scale1,scale2. (b) scale1,scale3. (c) scale1,scale4. (d) scale2,scale3. (e) scale2,scale4. (f) scale3,scale4.

leverages non-aligned multi-scale information, leading to more accurate and stable reconstruction as the sense ratios increase.

C. Ablation Study

Due to the space constraints, we use PM2.5, Humidity and P1 as the representative datasets. To assess the necessity of each core component in ST-NAI, we design the following variants:

1) *w/o Time-Continuous Modeling:* removing Time-Mamba and adopting a discrete-time processing scheme that corresponding real-world irregular sampling.
2) *w/o Spatial Non-Align Modeling:* removing the spatial non-alignment handling, using only single-scale input without geometric multi-scale alignment.
3) *w/o ST-Mamba:* replacing ST-Mamba with a vanilla Mamba backbone for spatiotemporal integration.
4) *w/o Cross-Mamba:* removing Cross-Mamba, disabling cross-scale information exchange.
5) *w/o Multi-scale loss:* removing the multi-scale loss and training with the finest scale loss only.

As shown in Table III, removing Time-Continuous Modeling causes a large performance drop, indicating that handling

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

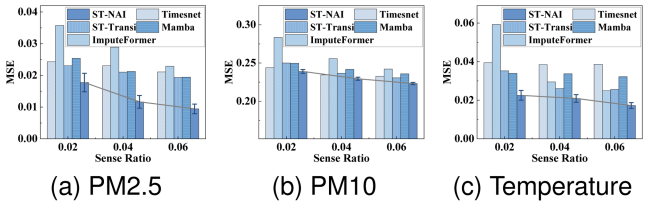


Fig. 11. Inference performance under time-continuous fairness comparison. (a) PM2.5. (b) PM10. (c) Temperature. (d) Humidity. (e) P1. (f) P2.

TABLE III
ABLATION ANALYSIS OF OUR METHOD

Method	Metric	Air-Quality	Weather	Traffic-Flow
w/o Time-Continuous	MSE	0.194	0.313	0.289
Modeling	MAE	0.343	0.437	0.363
w/o Spatial	MSE	0.013	0.042	0.237
Non-Align Modeling	MAE	0.075	0.149	0.299
w/o ST-Mamba	MSE	0.013	0.045	0.236
	MAE	**0.072**	0.155	0.298
w/o Cross-Mamba	MSE	0.012	0.048	0.233
	MAE	0.073	0.163	0.296
w/o Multi-scale loss	MSE	0.013	0.039	0.233
	MAE	0.075	0.144	0.294
ST-NAI (Our)	MSE	**0.012**	**0.037**	**0.229**
	MAE	**0.072**	**0.139**	**0.289**

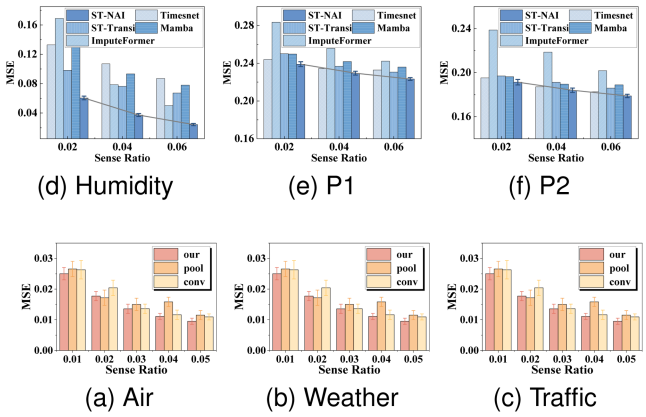


Fig. 12. Performance of Scale-Adjusted Resampling. (a) Air. (b) Weather. (c) Traffic.

irregular sampling is insufficient. The other variants also lead to consistent degradations, confirming that spatial non-alignment handling, spatiotemporal integration, cross-scale interaction, and multi-scale supervision all contribute to the final accuracy. Overall, the full ST-NAI achieves the best results, showing that these components are complementary.

We further conduct fine-grained ablation studies on key submodules, as detailed in the following subsections.

1) Scale-Adjusted Resampling: We conduct ablation studies on the spatial resampling module by replacing our scale-adjusted alignment with AdaptivePool (pool) and convolutional downsampling (conv). As shown in Fig. 12, both variants consistently yield higher MSE across sense ratios and scenarios,

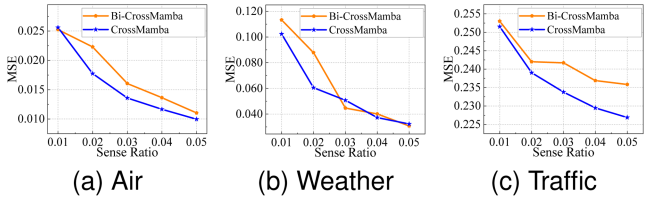


Fig. 13. Comparison between CrossMamba and Bi-CrossMamba. (a) Air. (b) Weather. (c) Traffic.

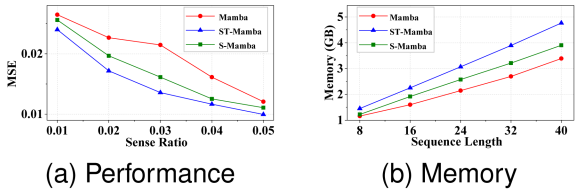


Fig. 14. Ablation of the ST-Mamba. (a) Performance. (b) Memory.

since they cannot guarantee geometric alignment when scales are non-divisible, leading to feature mismatch and information loss. In contrast, our scale-adjusted resampling explicitly aligns multi-scale grids before aggregation, and therefore achieves the best and most stable performance.

2) St-Mamba: We further evaluate the effectiveness and cost of ST-Mamba by comparing it with vanilla Mamba and an intermediate variant, S-Mamba, which removes the Spatial-Adjacent Scan. As shown in Fig. 14(a), ST-Mamba consistently achieves lower MSE, showing its advantage in capturing spatiotemporal patterns. Fig. 14(b) shows that the memory usage of all three models grows approximately linearly with the sequence length. ST-Mamba introduces moderate memory overhead, but it brings clear increase in performance.

3) Cross-Mamba: To examine whether bidirectional cross-scale interaction is beneficial, we compare CrossMamba with its bidirectional variant, Bi-CrossMamba, which further introduces a reverse fine-to-coarse interaction on top of the original coarse-to-fine design. As shown in Fig. 13, Bi-CrossMamba does not yield consistent improvements and is outperformed by CrossMamba in most cases. Since coarse-scale context has already been progressively injected into finer-scale representations in the preceding multi-scale pipeline, the additional reverse interaction brings limited benefit to the finest-scale target. Therefore, we adopt CrossMamba in the final model.

D. Parameter Sensitivity Analysis

1) Multi-Scale Loss C: We further evaluate the impact of the multi-scale weight coefficients C_s used in the loss function. Specifically, we examine three representative categories of weight settings: finest-dominant settings, where supervision is concentrated on the finest scale or the first few finer scales; uniform settings, where all scales receive equal weights; and unbalanced multi-scale settings, where the weights are distributed unevenly across scales. As shown in Table IV, when only the finest scale is emphasized, the model tends to produce relatively

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
SENSITIVITY ANALYSIS OF THE MULTI-SCALE LOSS WEIGHTS

C_1	C_2	C_3	C_4	Air-Quality	Weather	Traffic-Flow
1.0	0.0	0.0	0.0	0.0266	0.1077	0.2599
0.7	0.3	0.0	0.0	0.0259	0.1344	0.2737
0.6	0.3	0.1	0.0	0.0258	0.1038	0.2585
0.25	0.25	0.25	0.25	**0.0250**	0.1428	0.2624
0.5	0.3	0.15	0.05	0.0264	0.1252	0.2563
0.6	0.25	0.1	0.05	0.0268	0.1293	0.2604
0.2	0.4	0.3	0.1	0.0279	0.1094	0.2661
0.45	0.05	0.05	0.45	0.0258	0.1414	0.2623
0.4	**0.3**	**0.2**	**0.1**	0.0255	**0.1013**	**0.2521**

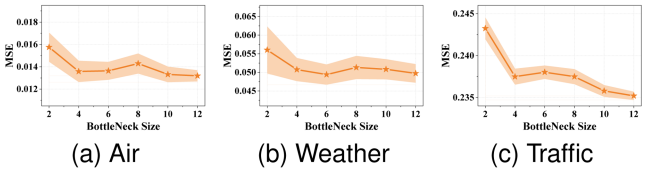


Fig. 15. Parameter Sensitivity Analysis of Bottleneck Size. (a) Air. (b) Weather. (c) Traffic.

higher errors, suggesting that auxiliary supervision from other scales is beneficial. In contrast, assigning equal weights to all scales does not consistently improve the performance, indicating that treating all scales equally may weaken the supervision of the finest-resolution target. In general, settings that place moderately larger weights on finer scales tend to yield more stable results. Considering both accuracy and robustness, we adopt the weight setting $\{0.4, 0.3, 0.2, 0.1\}$ in our experiments.

2) Bottleneck Size: We evaluate the sensitivity of the bottleneck size in Cross-Mamba. Fig. 15 shows the inference performance under different bottleneck sizes on three representative datasets. When the bottleneck size is very small (e.g., $n_b = 2$), the model tends to yield relatively higher errors, suggesting that limited bottleneck capacity may restrict the ability to capture cross-scale information. As the bottleneck size increases, the inference accuracy generally improves. However, further enlarging the bottleneck brings only marginal changes in performance. Considering both effectiveness and computational efficiency, we set $n_b = 4$ in our experiments.

E. Case Study

We have shown the importance of spatial nodes for air and traffic. As illustrated in Fig. 16, the importance scores for air quality data are notably concentrated in specific urban areas, likely indicating regions with significant environmental pressures or monitoring demands, such as industrial activity zones. Conversely, the importance scores for traffic-flow data are more uniformly distributed across the entire city, suggesting a widespread impact and connectivity of the urban traffic system. These differences suggest that our model can be responsive to different types of urban data.

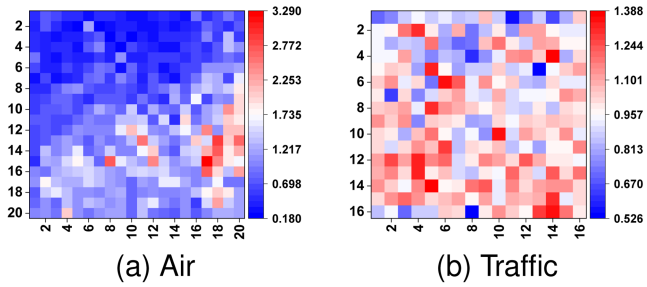


Fig. 16. Visualization of spatial importance. (a) Air. (b) Traffic.

TABLE V
COMPUTING OVERHEAD FOR MAIN METHODS

	Mamba	Transformer	ST-NAI
running time	22 s/epoch	246 s/epoch	48 s/epoch
memory cost	1520 MB	47624 MB	2550 MB

F. Computing Overhead

We compare the computational overhead of our model with Mamba and Transformer architectures. All runtime and memory measurements were conducted on a workstation equipped with an NVIDIA RTX 3090 GPU, an Intel i9-14900 K CPU, and 32 GB RAM, using PyTorch with CUDA acceleration. Unless otherwise specified, we use a batch size of 16, observed sequence length $L = 64$, target sequence length $T = 16$, and $S = 4$ spatial scales.

As shown in TABLE V, our model achieves significantly improved inference performance with computational costs slightly higher than Mamba, but substantially lower than those of the Transformer. This demonstrates our model's efficiency in balancing performance with computational demands.

VII. DISCUSSION

Although the spatial resampling remains linear under the assumption given in Theorem 4, real-world sensing scales may not always be strictly modeled as integer multiples of a common reference grid. In such cases, an approximate alignment strategy can be adopted by mapping all scales to a shared reference grid instead of strictly using the least common multiple. The reference grid can be chosen as the finest spatial resolution among the available scales or a predefined target resolution. This interpolation-based resampling effectively bounds the intermediate grid size while preserving most cross-scale information in practice.

VIII. CONCLUSION

In this paper, we investigate spatiotemporal non-aligned data inference in Sparse Mobile CrowdSensing, where observations are collected at irregular timestamps and across heterogeneous spatial scales. To address these challenges, we propose ST-NAI, a unified framework that models physical time gaps with Time-Mamba for time-continuous inference and resolves geometric mismatches across scales through scale-adjusted resampling.

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.

The framework further enhances spatiotemporal representation learning via ST-Mamba and enables effective information exchange across scales using Cross-Mamba. Extensive experiments on six real-world datasets demonstrate that ST-NAI consistently outperforms strong discrete and time-continuous baselines.

In future work, we plan to investigate more realistic sensing policies and adaptive spatial-scale selection under sensing budget constraints. We will also explore extensions of the framework to heterogeneous sensing environments involving diverse sensing modalities, device capabilities, and deployment settings to further improve the practicality and generalization of spatiotemporal non-aligned data inference.

REFERENCES

[1] 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.

[2] 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. Syst.*, 2010, pp. 312–317.

[3] H. Gong, B. Huang, B. Jia, and H. Dai, "Modeling power consumptions for multirotor UAVs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 6, pp. 7409–7422, Dec. 2023.

[4] H. Gong, B. Huang, and B. Jia, "Energy-efficient 3-D UAV ground node accessing using the minimum number of UAVs," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 12046–12060, Dec. 2024.

[5] 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.

[6] Y. Mostofi, "Compressive cooperative sensing and mapping in mobile networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 12, pp. 1769–1784, Dec. 2011.

[7] E. Wang, M. Zhang, Y. Yang, Y. Xu, and J. Wu, "Exploiting outlier value effects in sparse urban crowdsensing," in *Proc. IEEE/ACM Int. Symp. Qual. Service*, 2021, pp. 1–10.

[8] 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.

[9] 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 Conf. Comput. Commun.*, 2023, pp. 1–10.

[10] 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.

[11] S. Wang et al., "TimeMixer: Decomposable multiscale mixing for time series forecasting," in *Proc. 12th Int. Conf. Learn. Representations*, 2024, pp. 38626–38652.

[12] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 1–11.

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

[14] N. Marchang and R. Tripathi, "KNN-ST: Exploiting spatio-temporal correlation for missing data inference in environmental crowd sensing," *IEEE Sensors J.*, vol. 21, no. 3, pp. 3429–3436, 2020.

[15] G. Yang, P. Wen, Y. Liu, L. Kong, and Y. Liu, "Spatial-temporal data inference with graph attention neural networks in sparse mobile crowdsensing," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 5, pp. 4617–4626, Sep./Oct. 2024.

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

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

[18] 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.

[19] J. Zhang, Y. He, W. Chen, L.-D. Kuang, and B. Zheng, "CorrFormer: Context-aware tracking with cross-correlation and transformer," *Comput. Elect. Eng.*, vol. 114, 2024, Art. no. 109075.

[20] Y. Liu et al., "VMamba: Visual state space model," in *Proc. 38th Annu. Conf. Neural Inf. Process. Syst.*, 2024.

[21] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 62429–62442.

[22] L. Li, H. Wang, W. Zhang, and A. Coster, "STG-Mamba: Spatial-temporal graph learning via selective state space model," 2024, *arXiv:2403.12418*.

[23] Y. Zhu et al., "What to do next: Modeling user behaviors by time-LSTM," in *Proc. IJCAI*, Melbourne, VIC, 2017, vol. 17, pp. 3602–3608.

[24] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Proc. Adv. Neural Inf. Process. Syst.*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, vol. 31, pp. 1–13.

[25] Z. Yuan, X. Ban, Z. Zhang, X. Li, and H.-N. Dai, "ODE-RSSM: Learning stochastic recurrent state space model from irregularly sampled data," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 9, pp. 11060–11068.

[26] C. Coelho, M. F. P. Costa, and L. Ferrás, "Enhancing continuous time series modelling with a latent ODE-LSTM approach," *Appl. Math. Computation*, vol. 475, 2024, Art. no. 128727.

[27] Y. Zhang et al., "DIFFODE: Neural ode with differentiable hidden state for irregular time series analysis," in *Proc. IEEE 41st Int. Conf. Data Eng.*, 2025, pp. 1–14.

[28] A. Gu, "Modeling sequences with structured state spaces," Stanford University, 2023.

[29] A. Nagrani, S. Yang, A. Arnab, A. Jansen, C. Schmid, and C. Sun, "Attention bottlenecks for multimodal fusion," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 14200–14213.

[30] J. Wei et al., "Improved 1 km resolution PM 2.5 estimates across China using enhanced space–time extremely randomized trees," *Atmospheric Chem. Phys.*, vol. 20, no. 6, pp. 3273–3289, 2020.

[31] J. Wei et al., "The CinaHighPM$_{10}$ dataset: Generation, validation, and spatiotemporal variations from 2015 to 2019 across China," *Environ. Int.*, vol. 146, 2021, Art. no. 106290.

[32] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[33] 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.

[34] 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.

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

[36] W. Liu et al., "Non-aligned multi-scale data completion for sparse mobile crowdsensing," in *Proc. IEEE/ACM 33rd Int. Symp. Qual. Service (IWQoS)*, 2025, pp. 1–10.

[37] W. Liu et al., "Spatio-temporal pyramid-based multi-scale data completion in sparse crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 25, no. 1, pp. 1270–1286, Jan. 2026.

[38] H. Zhou et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 12, pp. 11106–11115.

[39] 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.

[40] B. Walker, A. D. McLeod, T. Qin, Y. Cheng, H. Li, and T. Lyons, "Log neural controlled differential equations: The lie brackets make a difference," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 1–23.

[41] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proc. Conf. Artif. Intell.*, 2023, vol. 37, pp. 4365–4373.



Hao Du received the BE degree in software engineering from Inner Mongolia University, Hohhot, China, in 2022 and began the graduation degree with Jilin University, Changchun, China, in 2022, and joined the combined masters and Ph.D. degree in 2024, where he is currently working toward the Ph.D. degree in computer science and technology. His research interests include mobile crowdsensing, spatiotemporal data processing, and multi-scale data processing.

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.



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



En Wang (Member, IEEE) received the BE degree in software engineering, and the ME and PhD degrees in computer science and technology from Jilin University, Changchun, China, in 2011, 2013, and 2016, respectively. He was 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. His research interests include mobile computing, crowd intelligence, and data mining.



Zhaoyang Sun (Graduate Student Member, IEEE) received the BSc degree in computer science and technology from Northeast Normal University, Changchun, China, in 2025. He has been working toward the MS degree in computer science and technology with Jilin University, Changchun, since 2025. His research interests include mobile crowdsensing and spatiotemporal data processing.



Chenyang Fu received the BE degree in computer science and technology from the Xi'an University of Technology, Xi'an, China, in 2025. He has been working toward the MS degree in software engineering with Jilin University, Changchun, China, since 2025. His research focuses on mobile crowdsensing.



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 research interests include 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 currently the director with the Center for Networked Computing and Laura H.Carnell professor with Temple University. He is the director with International Affairs, College of Science and Technology. He is also a chief scientist with China Telecom and the director with Cloud Computing Research Institute. He was the program director with National Science Foundation and a distinguished professor with Florida Atlantic University. His research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. He was a chair with the Department of Computer and Information Sciences from 2009 to 2016 and an associate vice provost with International Affairs from 2015 to 2017. Dr. Wu was an IEEE Computer Society distinguished visitor and ACM distinguished speaker. He is a China Computer Federation distinguished speaker.

Authorized licensed use limited to: Goa University. Downloaded on October 08,2026 at 08:56:03 UTC from IEEE Xplore. Restrictions apply.