Toward Time-Continuous Data Inference in Sparse Urban CrowdSensing

Hao Du[1,2], **Wenbin Liu**[1,2*], **Ziyu Sun**[1], **Haoyang Su**[1], **En Wang**[1,2*], **Yuanbo Xu**[1,2]

[1] College of Computer Science and Technology, Jilin University, China
[2] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, China
{duhao22,sunzy2121,suhy2121}@mails.jlu.edu.cn, {liuwenbin,wangen,yuanbox}@jlu.edu.cn,

Abstract

Sparse Urban CrowdSensing (Sparse UCS) is a practical paradigm for completing full sensing maps from limited observations. However, existing methods typically rely on a time-discrete assumption, where data is considered static within fixed intervals. This simplification introduces significant errors as real-world data changes continuously. To address this, we propose a framework for time-continuous data completion. Our approach, **Time**-Aware Mamba-based **D**eep **M**atrix **F**actorization (TIME-DMF), leverages the Mamba architecture as a powerful temporal encoder. Crucially, we enhance Mamba with a novel time-aware mechanism that explicitly incorporates the actual, often irregular, physical time intervals between observations into its state transitions. This allows our model to accurately capture true temporal dynamics and generate high-fidelity data for any queried moment in time through a query-generate mechanism. Extensive experiments on five diverse sensing tasks demonstrate that TIME-DMF significantly outperforms state-of-the-art methods, validating the superiority of the time-continuous paradigm for Sparse UCS.

Code — https://github.com/JLUDhhh/Time-DMF

Introduction

Mobile CrowdSensing (MCS) (Ganti, Ye, and Lei 2011; Wu and Wang 2023) has emerged as a transformative paradigm, leveraging the ubiquity of smart devices to create large-scale dynamic sensing networks. In practice, budget limitations and inaccessible areas often render MCS data incomplete. The Sparse UCS paradigm (Wang et al. 2016) addresses this challenge directly by employing inference strategies to reconstruct a complete sensing map from these partial observations. This has already shown great advantages in some practical applications, such as air quality monitoring (Liu et al. 2020; Feng et al. 2018), traffic control (Liu, Ong, and Chen 2020; Ali et al. 2021) and urban sensing (Calabrese, Ferrari, and Blondel 2014; Liu et al. 2019).

Data inference is the most essential part of Sparse UCS, yet most of existing works critically undermine its potential. As illustrated in Figure 1, existing methods study the data

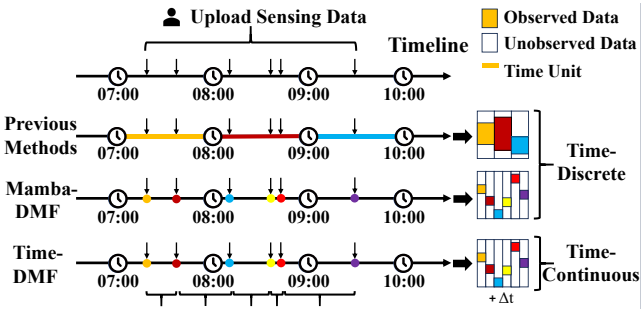


Figure 1: Time-discrete and time-continuous formulation.

inference problem from a time-discrete perspective (Wang et al. 2015, 2017; He and Shin 2018; Xie et al. 2019; Liu et al. 2022; Wang et al. 2022). The entire timeline of sensing data is typically discretized into fixed-length time units, and all observations arriving within the same unit are aggregated. Then, assuming intra-interval data stasis, they use data inference methods, such as compressive sensing (Aly, Basalamah, and Youssef 2016; He and Shin 2018) or matrix completion (Fan and Cheng 2018; Zhang and Chen 2019) to infer the missing data. However, this time-discrete assumption is routinely violated in practice, as the sensing data evolves continuously, resulting in significant errors in change-sensitive applications. This coarse-grained method inherently fails to resolve rapid changes, such as sharp fluctuations in environmental data during severe weather, thereby masking critical local dynamics.

Therefore, time-continuous data inference has become a pressing issue in Sparse UCS. In this paper, we shift from the traditional time-discrete paradigm to a more realistic time-continuous perspective. This fundamental change invalidates the common assumption of static data within fixed periods, precluding data aggregation and thus creating a challenge of extreme data sparsity. Consequently, our first challenge is to **develop a model robust to this sparsity**. Furthermore, the nature of real-time data collection results in irregular time intervals between observations. Our second challenge is to **effectively model and leverage this temporal irregularity**, which contains crucial dynamic information. Finally, discrete methods are inherently incapable of inferring data at arbitrary moments. Therefore, our third

*Corresponding authors.

challenge is to **achieve true continuous completion** by enabling on-demand inference for any point in time.

To address the challenges, we propose a progressive framework, starting with tackling extreme sparsity. By moving from coarse-grained aggregation to a "fine-grained" setting where each submission occupies a unique time step, we introduce Mamba-enabled Deep Matrix Factorization (Mamba-DMF). This model leverages the Mamba architecture (Gu and Dao 2023) as a temporal encoder to share information across time, effectively compensating for the lack of spatial context at each moment. However, Mamba-DMF remains fundamentally ordinal, processing events as an ordered sequence while ignoring the crucial physical time intervals between them. To exploit this temporal irregularity, we enhance our framework to create Time-Aware Mamba-DMF (TIME-DMF), whose core innovation is a mechanism that makes Mamba's state transitions explicitly sensitive to actual elapsed time. Finally, to achieve true continuous completion, we equip TIME-DMF with a Query-Generate (Q-G) strategy, transforming it from a static completion tool into a dynamic generative framework capable of providing on-demand inference for any arbitrary point in time.

Our work has the following contributions:

- We reformulate the problem of data inference from a time-continuous perspective in Sparse UCS. This new formulation abandons the flawed discrete-time assumption and addresses the fundamental challenges of extreme sparsity, temporal irregularity, and on-demand generation, bringing inference models closer to real-world dynamics.
- We propose TIME-DMF, a novel framework featuring a time-aware Mamba encoder. By introducing a dynamic timescale adaptation mechanism, TIME-DMF can explicitly and effectively leverage the irregular physical time intervals between observations, moving beyond the limitations of purely ordinal sequence processing.
- We design a Query-Generate (Q-G) strategy that works in synergy with TIME-DMF, transforming it into a generative neural process. This enables on-demand inference for any arbitrary moment in time, achieving true continuous data completion.
- Extensive experiments on four diverse, real-world datasets demonstrate the superiority of our time-continuous approach and the effectiveness of TIME-DMF, which significantly outperforms a wide range of state-of-the-art baselines.

Related Work

Our work intersects with two key research areas: data inference in Sparse UCS and continuous-time modeling.

Data Inference in Sparse UCS

Inferring complete data from partial observations is a central task in Sparse UCS (Wang et al. 2016). Methodologies for this task have evolved from relying on strong structural priors to leveraging data-driven deep learning models.

Early approaches were often built upon pre-defined assumptions about the data's structure. Compressive Sensing (CS), for instance, leverages signal sparsity in a transform domain to reconstruct the complete data (Aly, Basalamah, and Youssef 2016; He and Shin 2018). A more prevalent paradigm, Matrix Completion (MC), typically assumes that the underlying spatiotemporal data matrix is low-rank, enabling recovery from a small subset of entries (Fan and Cheng 2018; Zhang and Chen 2019). While effective in data-scarce scenarios, the performance of these methods is often constrained by their strong structural priors, which may not adequately capture complex real-world dynamics.

To overcome the limitations of these prior-based methods, recent research has shifted towards deep learning. Transformers, in particular, have demonstrated state-of-the-art performance by learning intricate spatiotemporal patterns from large, complete datasets (Liu et al. 2023; Chen et al. 2021). However, the main limitation of these powerful, data-driven models is their reliance on extensive historical data, a luxury often unavailable in practical Sparse UCS applications.

Despite their different underlying philosophies, both paradigms are almost universally built upon a flawed time-discrete assumption, aggregating data into fixed slots (Wang et al. 2017). This process fundamentally ignores the continuous nature of real-world phenomena, erodes temporal fidelity, and discards crucial information encoded in irregular time intervals.

Continuous-Time Modeling

Our work directly confronts this blind spot by focusing on temporal granularity. While fine-grained spatial modeling often requires costly sensor deployment (Thepvilojanapong, Ono, and Tobe 2010), continuous-time modeling has been less explored in Sparse UCS.

Relevant prior work includes continuous extensions to classical models like ARMA (Brockwell 2001) and event-based state-space approaches (Higuchi 1988; Kidger et al. 2020). More directly, Zhu $et\ al.$ (Zhu et al. 2017) pioneered the use of time gates in deep learning to handle unequal intervals in recommendation systems. The advent of selective State Space Models like Mamba (Gu and Dao 2023) offers a new, powerful mechanism, as its ability to dynamically adapt its internal state transitions based on input content is inherently suitable for handling varying time intervals. Drawing from these insights, we propose our framework for time-continuous completion in Sparse UCS.

System Model and Problem Formulation

In this section, we first establish the system model, beginning with the traditional time-discrete formulation and then introducing our novel time-continuous approach. Subsequently, we provide a formal problem formulation.

System Model

In Sparse UCS tasks, the sensing map covers N regions over M irregular time slices. At moment t, a submission is represented by a one-hot position vector $c_{(t)} \in \mathbb{R}^N$ (with the i-th element as 1 if data comes from the i-th sub-region) and

a data vector $y'_{(t)} \in \mathbb{R}^N$ (with the i-th element as the submitted value, and others set to meaningless values like 0 or negative).

For M submissions, users sense partial locations, yielding the position matrix $C \in \mathbb{R}^{N\times M}$ and observed data matrix $Y' \in \mathbb{R}^{N\times M}$:

$$C = [c_1^{\mathrm{T}}, c_2^{\mathrm{T}}, \ldots, c_M^{\mathrm{T}}], \tag{1}$$

$$Y' = [y_1'^{\mathrm{T}}, y_2'^{\mathrm{T}}, \ldots, y_M'^{\mathrm{T}}]. \tag{2}$$

The ground truth $Y = [y_1^{\mathrm{T}}, y_2^{\mathrm{T}}, \ldots, y_M^{\mathrm{T}}]$ gives:

$$Y' = Y \odot C, \tag{3}$$

where $\odot$ is the Hadamard product.

Unlike traditional methods aggregating into discrete units, our approach uses precise temporal information in T, dividing the problem into two subtasks.

The first finds mapping $f(\cdot)$ to complete Y' leveraging T, similar to time-discrete scenarios:

$$\hat{Y} = f(Y') \approx Y. \tag{4}$$

The second identifies model $g(\cdot)$ for inference $\hat{y} \in \mathbb{R}^N$ at any t:

$$\hat{y} = g(Y', t), \quad t \in (t_0, t_M). \tag{5}$$

B. Problem Formulation

Problem : Given sparse sensed data Y' and time vector T, we aim to achieve the following two objectives:

- Identify a mapping $f(\cdot)$ to complete all the unsensed data in the matrix Y'. The mapping $f(\cdot)$ should adequately consider the high sparsity of Y' and the temporal information in T.
- Identify a model $g(\cdot)$ to accomplish the completion at any given time t'. $y^{(t')\prime}$ can be a column in Y' or not.

In this process, the mean square error is used to measure the quality of the completed and generated data. The following value should be minimized:

$$\epsilon(Y, Y') = \sum_i^N \sum_j^M |Y_{ij} - Y'_{ij}|. \tag{6}$$

Methodology

Overall Framework

Our proposed framework, TIME-DMF, addresses three challenges in time-continuous data completion: extreme sparsity, temporal irregularity, and on-demand generation. It builds on DMF principles, evolving into a time-aware system. To tackle sparsity, we first introduce Mamba-DMF, leveraging Mamba as a temporal encoder to share information across time. TIME-DMF then extends it for time-continuity, incorporating a time-aware mechanism that makes state transitions sensitive to physical intervals. In Figure 2, learnable tensors process through Mamba for temporal correlations, followed by DMF for low-rank relations in completion; Figure 3 details the Time-Aware Mamba. This achieves a mapping from sparse, asynchronous observations to a continuous spatiotemporal representation via Q-G for on-demand inference.

The model takes as input the primary low-rank representation of the data, denoted as $Z_{prim} = [X_1, X_2, \ldots, X_M]$, where each X_t is a learnable embedding for a submission at time *t*. Crucially, it also takes the vector of precise physical time intervals between submissions, $\tilde{\Delta}T_{raw}$, as an additional input. The TIME-DMF framework consists of two primary components:

A Time-Aware Mamba Encoder. This is the central innovation of our work. Unlike standard sequence models that are merely ordinal, our encoder is explicitly designed to be sensitive to the physical passage of time. It processes the sequence of primary low-rank vectors Z_{prim} while dynamically incorporating the information from the time interval vector $\tilde{\Delta}T_{raw}$. This allows the encoder to learn nuanced, physically-grounded temporal dependencies. The output of this stage is a sequence of context-aware, encoded low-rank representations Z_{enc}.

A DMF-like Decoder. Following the encoder, we employ a non-linear projection function, similar to the decoder in the standard DMF framework. This decoder takes the encoded, context-aware representations Z_{enc} and maps them back to the high-dimensional data space, producing the final completed data matrix $\hat{Y}$.

$$\hat{Y} = f(Z_{enc}). \tag{7}$$

By jointly training the encoder and decoder, TIME-DMF learns to effectively leverage both the content of the submissions and their precise temporal context. This integrated design enables the model to perform robustly even with extremely sparse and irregularly sampled data, forming the basis for our subsequent time-continuous inference capabilities.

Mamba-Enabled Deep Matrix Factorization (Mamba-DMF)

To confront the challenge of extreme data sparsity, a natural solution is to enable information sharing across time steps. This requires a powerful sequence model to encode the temporal context within a proven non-linear factorization framework like DMF. To this end, we propose Mamba-enabled Deep Matrix Factorization (Mamba-DMF).

1) Foundational Components

Deep Matrix Factorization (DMF). As a foundational framework for the following works, DMF extends traditional matrix factorization by learning a non-linear mapping $f(\cdot)$ from a low-rank representation Z to the full data space Y. This is achieved by fitting the input vectors Z and the parameters of a deep neural network (DNN), which represents the function $f(\cdot)$. The parameters of both the latent representation Z and the DNN are optimized jointly during training. (See Appendix for detailed descriptions and visual instruction). This structure allows it to capture complex, non-linear spatiotemporal correlations:

$$Y = f(Z). \tag{8}$$

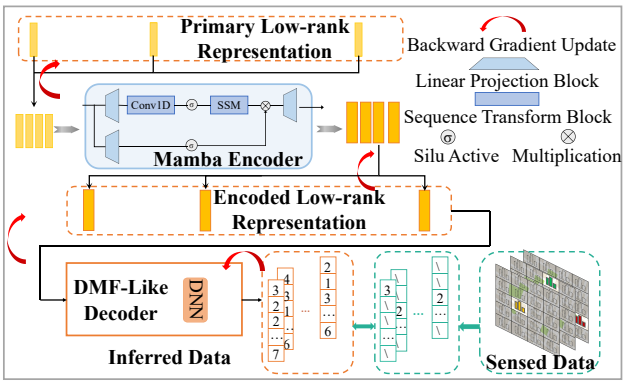


Figure 2: The inner structure of Mamba-DMF.

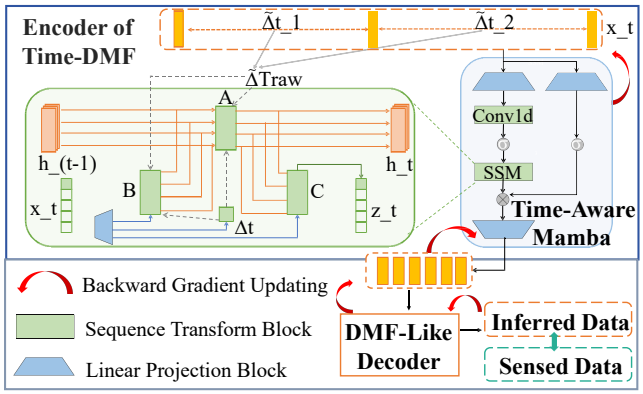


Figure 3: The inner structure of TIME-DMF.

However, in its original form, DMF learns the low-rank vector for each time step independently, making it unable to leverage temporal correlations to combat extreme sparsity.

Mamba Architecture. Mamba (Gu and Dao 2023) is a selective State Space Model (SSM) that excels at modeling long sequential data with linear-time complexity. Its core strength lies in its ability to selectively process information and capture context-dependent, long-range dependencies, making it an ideal candidate for a temporal encoder. (See Appendix for detailed description).

2) The Mamba-DMF Framework

The central idea of Mamba-DMF is to address the limitation of the standard DMF by explicitly modeling the temporal dependencies between the low-rank representations of different time steps. Instead of learning independent low-rank vectors, we treat them as a sequence and use Mamba to encode this sequence. As shown in Figure 2, the framework operates as follows:

Primary Low-Rank Representation. We begin with a sequence of learnable primary low-rank vectors $Z_{prim} = [X_1, X_2, \ldots, X_M]$. Each X_t serves as a learnable parameter representing the initial embedding for the t-th time step.

Mamba Encoder. The sequence Z_{prim} is fed into a Mamba Encoder. This encoder processes the entire sequence, allowing information to propagate across all time steps. It captures the temporal correlations and produces a sequence of context-aware, encoded low-rank representations Z_{enc}:

$$Z_{enc} = \text{MambaEncoder}(Z_{prim}), \tag{9}$$

DMF Decoder. Finally, the encoded sequence Z_{enc}, now imbued with rich temporal context extracted by Mamba, is subsequently fed into the downstream DMF decoder:

$$\hat{Y} = f(Z_{enc}). \tag{10}$$

Here, $f(\cdot)$ represents the non-linear projection function of the DMF decoder.

By structuring the problem this way, Mamba-DMF transforms the task from independent vector completion to sequence-to-sequence modeling. The Mamba encoder effectively shares information across time, allowing the model to make robust inferences for a given time step by leveraging the context of the entire timeline. This approach proves significantly more effective than the standard DMF in extremely sparse scenarios. All components of the Mamba-DMF model are jointly optimized.

Time-Aware Mamba-DMF (TIME-DMF)

While Mamba-DMF effectively addresses sparsity by modeling temporal sequences, it inherits a fundamental limitation from standard sequence models: it is ordinal, not truly temporal. It processes events based on their order but remains oblivious to the actual, non-uniform physical time intervals that separate them. This insensitivity to real-world time progression motivates our core innovation: Time-Aware Mamba Encoder for Continuous Dynamics.

The key to achieving time-awareness is to make the Mamba encoder's internal state transitions explicitly sensitive to the passage of physical time. (See Appendix for details). We achieve this by dynamically modulating Mamba's timescale parameter, which we denote as $\tilde{\Delta}_t$, based on the actual time intervals between submissions.

Physical Time Interval Calculation. Given a sequence of timestamps $T = [t_1, t_2, \ldots, t_M]$, where M is the total number of time steps in the current input sequence, we first compute the sequence of raw physical time intervals, $\tilde{\Delta}T_{raw} = [\tilde{\Delta}t_1, \tilde{\Delta}t_2, \ldots, \tilde{\Delta}t_M]$. For each time step k from 1 to M, the interval $\tilde{\Delta}t_k$ is calculated as $\tilde{\Delta}t_k = t_k - t_{k-1}$. For the initial step t_1, $\tilde{\Delta}t_1$ is conventionally set to zero, representing no elapsed time before the first observation in the sequence. This raw sequence undergoes suitable preprocessing to ensure numerical stability for model input.

Adaptive Timescale Fusion. The standard Mamba model generates its timescale parameter $\tilde{\Delta}_t$ based on the input content x_t. Our key insight is to fuse this content-driven timescale with the preprocessed physical time interval sequence $\tilde{\Delta}T_{raw}$. As illustrated in Figure 3, we introduce a learnable gating mechanism to adaptively combine these two sources of temporal information. First, the preprocessed physical time interval sequence $\tilde{\Delta}T_{raw}$ is transformed by a

Dataset	Sensed	GP	KNN-S	MC	DMF	STformer	iTransformer	Autoformer	Mamba-DMF
Sensor-Scope	1/57	4.3	4.2	3.9	4.7	6.6	6.1	5.3	**2.5** $\pm$ 0.36
	2/57	4.1	4.1	2.5	2.4	6.5	5.8	5.2	**2.0** $\pm$ 0.09
	3/57	3.6	3.2	2.5	2.0	5.6	5.0	5.0	**1.8** $\pm$ 0.06
	4/57	3.3	3.3	1.7	1.9	5.4	4.7	4.9	**1.7** $\pm$ 0.07
	5/57	3.2	3.2	1.6	1.7	5.0	4.1	4.9	**1.5** $\pm$ 0.05
U-AIR	1/36	53.0	55.8	56.5	60.7	51.4	74.5	90.4	**43.6** $\pm$ 1.86
	2/36	54.2	44.6	44.4	46.5	47.9	67.8	89.3	**38.5** $\pm$ 2.24
	3/36	53.0	41.6	38.9	39.3	35.8	62.2	88.5	**34.4** $\pm$ 4.59
	4/36	52.5	39.4	36.9	38.6	34.3	57.2	88.1	**34.1** $\pm$ 1.26
	5/36	50.6	58.2	34.1	37.7	33.5	53.3	87.2	**28.8** $\pm$ 1.53
Highways England	1/15	66.5	70.5	66.1	40.7	30.7	79.8	52.0	**22.9** $\pm$ 0.8
	2/15	56.2	60.3	49.6	31.6	19.2	58.3	51.7	**18.1** $\pm$ 0.48
	3/15	50.7	53.8	41.6	28.5	17.6	43.5	51.2	**17.5** $\pm$ 0.23
	4/15	45.9	49.9	36.9	26.9	16.2	31.7	42.4	**15.8** $\pm$ 0.22
	5/15	42.5	52.4	29.8	25.9	13.0	26.2	37.2	**11.7** $\pm$ 0.23
TaxiSpeed	1/30	39028.1	38252.8	31450.2	10474.5	8778.2	8876.7	11921.1	**7632.0** $\pm$ 91
	2/30	39990.3	32182.1	28295.3	9859.5	7831.3	8691.6	11876.0	**7493.1** $\pm$ 70
	3/30	38637.5	29491.2	24141.7	9220.7	7683.9	8414.7	9139.1	**7361.5** $\pm$ 82
	4/30	36617.3	28337.3	22709.9	7417.5	7428.0	8277.5	8984.9	**7241.1** $\pm$ 86
	5/30	35991.2	26787.2	21344.1	7012.0	6713.5	8139.8	8803.8	**6472.2** $\pm$ 71

Table 1: Full RMSE results under different sparsity on four datasets.

dedicated projection network:

$$\tilde{\Delta}T_{proj} = \text{Projection}(\tilde{\Delta}T_{raw}). \tag{11}$$

After that, a gate G is then computed using both the projected physical intervals $\tilde{\Delta}T_{proj}$ and the original content-driven timescale $\tilde{\Delta}_t$:

$$G = \sigma(\text{GateNet}([\tilde{\Delta}T_{proj}, \tilde{\Delta}_t])), \tag{12}$$

where $GateNet(.)$ is an MLP network. The final timescale parameter is denoted as:

$$\Delta_f = G \odot \tilde{\Delta}T_{proj} + (1 - G) \odot \tilde{\Delta}_t. \tag{13}$$

where $\odot$ represents element-wise multiplication.

Time-Aware State Transitions. With the fused timescale Δ_f updating equations, the state transition matrices A and B, which govern the evolution of Mamba's latent state, are now functions of both the data content and the real-world time flow:

$$\begin{aligned} \bar{\mathbf{A}} &= \exp(\Delta_f \mathbf{A}), \\ \bar{\mathbf{B}} &= (\Delta_f \mathbf{A})^{-1}(\exp(\Delta_f \mathbf{A}) - \mathbf{I}) \cdot \Delta_f \mathbf{B}. \\ h_t &= \bar{\mathbf{A}} h_{t-1} + \bar{\mathbf{B}} x_t, y_t = \mathbf{C} h_t. \end{aligned} \tag{14}$$

By introducing this mechanism, our Time-Aware Mamba Encoder can dynamically adapt its behavior. For long time gaps, it can learn to expect greater state changes, while for short intervals, it can enforce smoother transitions. This adaptation is achieved without altering the underlying parallelizable structure of Mamba's Selective Scan algorithm, thus preserving its computational efficiency. The resulting time-aware encoder, combined with the DMF-like decoder, forms the complete TIME-DMF framework, which is trained end-to-end.

Algorithm 1: Query-Generate (Q-G) Strategy

Input: Sparse data Y', time vector T, query time t
Output: Inferred vector $\hat{y}$ at time t

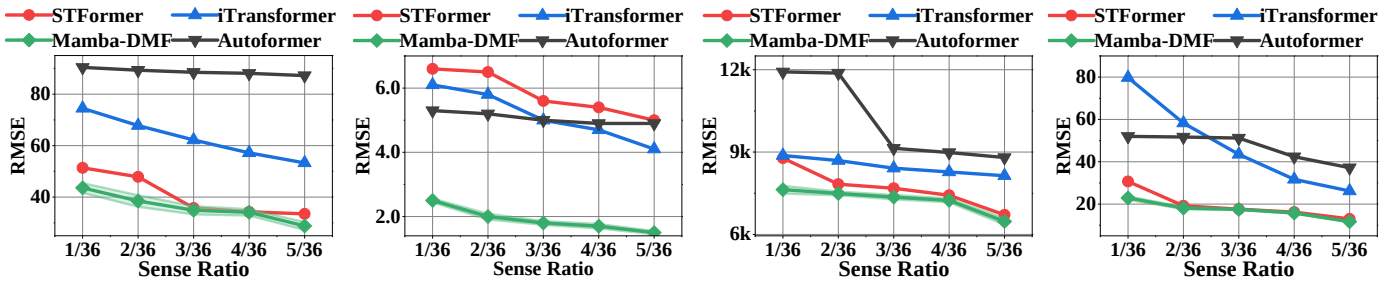
1: Generate random vector $X_q \sim \mathcal{N}(0, I)$.
2: Insert X_q into low-rank matrix Z' at position for t.
3: Update raw time intervals ΔT_{raw} based on new t in T.
4: Compute $\hat{Y} = \text{TIME-DMF}(Y', Z', \Delta T_{\text{raw}})$.
5: Extract $\hat{y}$ as the column of $\hat{Y}$ corresponding to t.
6: **return** $\hat{y}$

Query-Generate (Q-G) Strategy

While our TIME-DMF model can infer missing data for observed time points, a truly continuous approach must handle the infinite moments on a timeline, a task impossible for standard completion strategies. To overcome this, we propose the Query-Generate (Q-G) strategy, an inference-time procedure that enables on-demand data generation for any arbitrary moment. The strategy leverages the generative na-

Dataset	GP	KNN-S	MC	DMF	STformer	iTransformer	Autoformer	TIME-DMF
Sensor-Scope	14.0	40.4	49.7	50.6	13.7	14.8	15.2	**12.6**
U-AIR	105.7	93.0	96.9	103.3	**80.2**	84.4	90.2	80.8
TaxiSpeed	33213.7	26283.4	23232.2	13129.9	10068.7	10337.1	10748.0	**8822.4**
Hishways England A	66.5	70.5	66.1	71.5	27.3	47.6	42.6	**22.9**
Hishways England B	146.7	129.6	89.1	82.9	47.0	79.5	54.0	**39.6**
Hishways England C	207.7	182.0	109.6	124.1	50.8	100.2	103.4	**42.5**

Table 2: Direct comparison of time-continuous and time-discrete methods.



(a) U-AIR (b) Sensor-Scope (c) TaxiSpeed (d) Highways England

Figure 4: Partial RMSE results of different data sparsity under different datasets.

ture of our framework: for a query at a new time point t_q, we insert a randomly initialized placeholder vector X_q into the primary low-rank sequence Z_{prim} and correspondingly update the physical time interval sequence $\tilde{\Delta}T_{raw}$. We then perform a full forward pass through the pre-trained TIME-DMF model to dynamically generate a context-aware result. A key aspect is that this process does not affect the model's learned parameters, as the placeholder X_q is isolated from any gradient backpropagation during training or fine-tuning.

Experiments

Experimental Setup

In this section, we introduce the datasets and the comparison methods.

1) Datasets: We evaluate our models on four diverse real-world datasets: U-Air (Zheng, Liu, and Hsieh 2013), Sensor-Scope (Ingelrest et al. 2010), TaxiSpeed (Shang et al. 2014) and Highways England (HE) (Service 2022). (See Appendix for detailed descriptions).

2) Baselines: We compare our method with seven Sparse-supervised models for completion task: MC, KNN-S, GP, DMF (Wang et al. 2020), STformer (Wang et al. 2023), iTransformer (Liu et al. 2023) and Autoformer (Chen et al. 2021), as well as three predictive models for generative task: LINEAR, WNN and NAR. (See Appendix for detailed descriptions on each baseline).

Performance Comparison

In typical sparse data completion tasks, sensing rates range from 20% to 80%, providing ample spatiotemporal information. However, fine-grained time-continuous completion involves matrices with sensing ratios of $1/n$ per column, posing challenges for traditional methods. To validate our hypothesis, we first demonstrate Mamba-based models' effectiveness on extremely sparse data, then highlight the time-continuous paradigm's clear advantages.

Table 1 shows completion results on extremely sparse data (1–5 observations per column). Mamba-DMF outperforms all baselines, including Transformer variants. Figure 4 visualizes this across varying sensing ratios, revealing consistent and growing advantages as data increases. This confirms Mamba's inductive bias excels at capturing long-range dependencies in sparse sequences.

We further compare time-continuous TIME-DMF against time-discrete methods. As in Table 2, TIME-DMF significantly surpasses all discrete competitors across datasets. This underscores that avoiding aggregation yields superior temporal precision, outweighing sparsity challenges, and establishes the time-continuous approach as fundamentally better for real-world Sparse UCS.

Ablation Study

To verify our time-aware mechanism's effectiveness, we compare TIME-DMF with Mamba-DMF. Since public datasets often feature uniform sampling, we simulate irregularity by randomly deleting columns from the full Highways England matrix, then applying standard masking. As in Table 3, TIME-DMF consistently outperforms Mamba-DMF, with the gap widening at higher deletion ratios. Both degrade with increased unevenness, but TIME-DMF shows greater robustness: its timescale dynamically adapts to physical intervals, unlike Mamba-DMF's content-only approach. This allows it to better adapt its state transitions to the actual

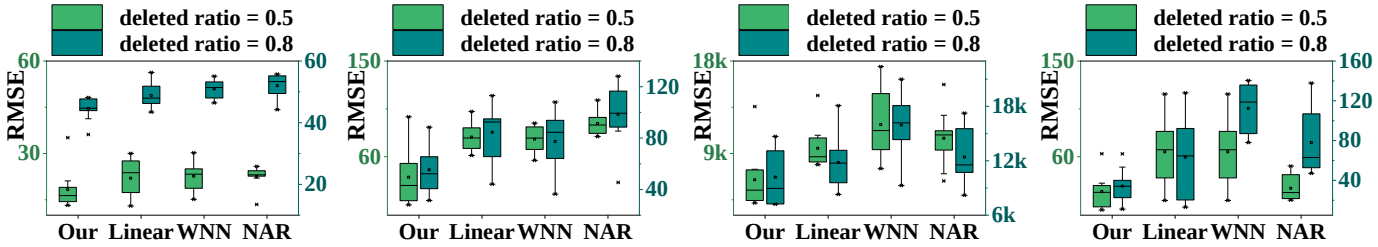
Dataset	Method	Slices Deleted Ratio				
		0.5	0.6	0.7	0.8	0.9
1month	Mamba-DMF	30.4	34.0	38.4	51.0	75.2
	TIME-DMF	**29.2**	**29.5**	**31.4**	**32.5**	**42.8**
2months	Mamba-DMF	**31.5**	34.7	38.5	44.2	58.7
	TIME-DMF	31.8	**34.6**	**36.8**	**37.5**	**40.7**
3months	Mamba-DMF	**34.1**	36.5	40.8	48.2	76.6
	TIME-DMF	34.4	**36.0**	**37.2**	**38.0**	**41.2**

Highways England for 2021

Dataset	Method	Slices Deleted Ratio				
		0.5	0.6	0.7	0.8	0.9
2weeks	Mamba-DMF	**29.9**	32.5	37.1	41.7	56.2
	TIME-DMF	30.3	**31.4**	**32.4**	**34.3**	**37.3**
1month	Mamba-DMF	30.4	33.9	38.8	49.2	81.4
	TIME-DMF	**29.3**	**31.6**	**32.1**	**33.4**	**35.5**
2months	Mamba-DMF	**59.2**	**63.7**	69.8	78.5	115.3
	TIME-DMF	59.9	64.9	**66.1**	**66.4**	**68.7**

Highways England for 2022

Table 3: RMSE results of the ablation study of time gates.



(a) U-AIR (b) Sensor-Scope (c) TaxiSpeed (d) Highways England

Figure 5: RMSE of different data unevenness under different datasets.

Model	Sparse-Supervised Only	Parameter Count	Training Time / Epoch
Time-DMF	✓	**12k**	**1.8ms**
iTransformer	×	51k	58ms
Autoformer	×	56k	66ms
STformer	✓	11M	400ms

Table 4: Training condition, speed and parameter count.

temporal spacing of the data, directly proving the effectiveness of our time-aware enhancements.

Demonstration of Generative Capability

Our time-continuous paradigm enables data generation at any moment, beyond completing fixed observations. We evaluate TIME-DMF's generative ability via single-moment generation tasks, comparing it against predictive models (configured for single-step prediction) since traditional completion methods lack this feature.

To test under temporal irregularity, we create datasets via random column deletion and masking. Given single-point instability, we run multiple random constructions and show aggregated box plots in Figure 5. The results show that TIME-DMF achieves a superior overall generation accuracy, evidenced by the lower median error across most datasets. The box plots also reveal that as temporal unevenness increases, the performance of all models becomes more variable, but TIME-DMF consistently maintains its advantage.

Model Parameter Comparison

Transformers, popular for time-series modeling, often require complete historical data, which is unavailable in sparse UCS scenarios. Our Mamba-based TIME-DMF is more efficient and suitable, excelling in data-scarce conditions. As shown in Table 4, it features fewer parameters and faster per-epoch training. These advantages make it ideal for resource-constrained environments, like on-device UCS deployments.

Conclusion

In this paper, we challenge the prevailing, yet flawed, assumption in Sparse UCS that data remains static within discrete time periods. To address this, we propose Time-Aware Mamba-DMF (TIME-DMF), a novel framework for time-continuous completion. Building upon a DMF foundation, TIME-DMF incorporates an enhanced Mamba-based temporal encoder. The core of our innovation lies in making this encoder time-aware: it dynamically modulates Mamba's internal timescale parameters based on the actual physical time intervals between observations. This mechanism, leveraging Mamba's selective state-space architecture, enables TIME-DMF to effectively capture complex temporal dependencies from non-uniformly sampled data. In cooperation with a Query-Generate (Q-G) strategy, our framework provides on-demand, generative responses for any arbitrary moment. Extensive experiments on real-world datasets validate the effectiveness of our approach.

Acknowledgments

This work is supported in part by National Key R&D Program of China under Grant Nos. 2021ZD0112501 and 2021ZD0112502, and National Natural Science Foundation of China under Grant Nos. 62472194 and 62272193, and Jilin Science and Technology Research Project 20250102220JC.

References

Ali, A.; Qureshi, M. A.; Shiraz, M.; and Shamim, A. 2021. Mobile crowd sensing based dynamic traffic efficiency framework for urban traffic congestion control. *Sustainable Computing: Informatics and Systems*, 32: 100608.

Aly, H.; Basalamah, A.; and Youssef, M. 2016. Automatic rich map semantics identification through smartphone-based crowd-sensing. *IEEE Transactions on Mobile Computing*, 16(10): 2712–2725.

Brockwell, P. J. 2001. Continuous-time ARMA processes. *Handbook of statistics*, 19: 249–276.

Calabrese, F.; Ferrari, L.; and Blondel, V. D. 2014. Urban sensing using mobile phone network data: a survey of research. *Acm computing surveys (csur)*, 47(2): 1–20.

Chen, M.; Peng, H.; Fu, J.; and Ling, H. 2021. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 12270–12280.

Fan, J.; and Cheng, J. 2018. Matrix completion by deep matrix factorization. *Neural Networks*, 98: 34–41.

Feng, C.; Tian, Y.; Gong, X.; Que, X.; and Wang, W. 2018. MCS-RF: mobile crowdsensing–based air quality estimation with random forest. *International Journal of Distributed Sensor Networks*, 14(10): 1550147718804702.

Ganti, R. K.; Ye, F.; and Lei, H. 2011. Mobile crowdsensing: current state and future challenges. *IEEE communications Magazine*, 49(11): 32–39.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

He, S.; and Shin, K. G. 2018. Steering crowdsourced signal map construction via Bayesian compressive sensing. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, 1016–1024. IEEE.

Higuchi, T. 1988. Approach to an irregular time series on the basis of the fractal theory. *Physica D: Nonlinear Phenomena*, 31(2): 277–283.

Ingelrest, F.; Barrenetxea, G.; Schaefer, G.; Vetterli, M.; Couach, O.; and Parlange, M. 2010. Sensorscope: Application-specific sensor network for environmental monitoring. *ACM Transactions on Sensor Networks (TOSN)*, 6(2): 1–32.

Kidger, P.; Morrill, J.; Foster, J.; and Lyons, T. 2020. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33: 6696–6707.

Liu, J.; Ong, G. P.; and Chen, X. 2020. GraphSAGE-based traffic speed forecasting for segment network with sparse data. *IEEE Transactions on Intelligent Transportation Systems*, 23(3): 1755–1766.

Liu, W.; Wang, E.; Yang, Y.; and Wu, J. 2022. Worker Selection Towards Data Completion for Online Sparse Crowdsensing. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, 1509–1518. IEEE.

Liu, W.; Yang, Y.; Wang, E.; Wang, L.; Zeghlache, D.; and Zhang, D. 2019. Multi-dimensional urban sensing in sparse mobile crowdsensing. *IEEE Access*, 7: 82066–82079.

Liu, W.; Yang, Y.; Wang, E.; and Wu, J. 2020. Fine-grained urban prediction via sparse mobile crowdsensing. In *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 265–273. IEEE.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

Service, G. D. 2022. Highways England. data.gov.uk/dataset/highways-england-network-journey-time-and-traffic-flow-data. 2022.12.

Shang, J.; Zheng, Y.; Tong, W.; Chang, E.; and Yu, Y. 2014. Inferring gas consumption and pollution emission of vehicles throughout a city. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1027–1036.

Thepvilojanapong, N.; Ono, T.; and Tobe, Y. 2010. A deployment of fine-grained sensor network and empirical analysis of urban temperature. *Sensors*, 10(3): 2217–2241.

Wang, E.; Liu, W.; Liu, W.; Xiang, C.; Yang, B.; and Yang, Y. 2023. Spatiotemporal Transformer for Data Inference and Long Prediction in Sparse Mobile CrowdSensing. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, 1–10. IEEE.

Wang, E.; Zhang, M.; Cheng, X.; Yang, Y.; Liu, W.; Yu, H.; Wang, L.; and Zhang, J. 2020. Deep learning-enabled sparse industrial crowdsensing and prediction. *IEEE Transactions on Industrial Informatics*, 17(9): 6170–6181.

Wang, E.; Zhang, M.; Liu, W.; Xiong, H.; Yang, B.; Yang, Y.; and Wu, J. 2022. Outlier-Concerned Data Completion Exploiting Intra-and Inter-Data Correlations in Sparse CrowdSensing. *IEEE/ACM Transactions on Networking*.

Wang, L.; Zhang, D.; Pathak, A.; Chen, C.; Xiong, H.; Yang, D.; and Wang, Y. 2015. CCS-TA: Quality-guaranteed online task allocation in compressive crowdsensing. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, 683–694.

Wang, L.; Zhang, D.; Wang, Y.; Chen, C.; Han, X.; and M'hamed, A. 2016. Sparse mobile crowdsensing: challenges and opportunities. *IEEE Communications Magazine*, 54(7): 161–167.

Wang, L.; Zhang, D.; Yang, D.; Pathak, A.; Chen, C.; Han, X.; Xiong, H.; and Wang, Y. 2017. SPACE-TA: Cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(2): 1–28.

Wu, J.; and Wang, E. 2023. *Mobile Crowdsourcing From Theory to Practice*. Springer.

Xie, K.; Li, X.; Wang, X.; Xie, G.; Wen, J.; and Zhang, D. 2019. Active sparse mobile crowd sensing based on matrix completion. In *Proceedings of the 2019 International Conference on Management of Data*, 195–210.

Zhang, M.; and Chen, Y. 2019. Inductive matrix completion based on graph neural networks. *arXiv preprint arXiv:1904.12058*.

Zheng, Y.; Liu, F.; and Hsieh, H.-P. 2013. U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1436–1444.

Zhu, Y.; Li, H.; Liao, Y.; Wang, B.; Guan, Z.; Liu, H.; and Cai, D. 2017. What to Do Next: Modeling User Behaviors by Time-LSTM. In *IJCAI*, volume 17, 3602–3608.