AS-STG: Spatio-Temporal Graph Learning with Active Sampling for Dynamic IoT Sensing

Yaxin Mei[1,2], Jiandian Zeng[1], Huiling Qin[1,*], Guangxue Zhang[1], Xi Zheng[3], Qin Liu[4], and Tian Wang[1]

[1]Institute of Artificial Intelligence and Future Networks, Beijing Normal University, Zhuhai, China
[2]School of Artificial Intelligence, Beijing Normal University, Beijing, China
[3]School of Computing, Macquarie University, NSW, Australia
[4]College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
meiyax@163.com, zengjiandian@foxmail.com, orekinana@gmail.com, guangxue_zhang@bnu.edu.cn,
james.zheng@mq.edu.au, gracelq628@hnu.edu.cn, tianwang@bnu.edu.cn

***Abstract*—Efficient sensing is critical for Internet of Things (IoT) applications, such as environmental monitoring and traffic management, where high quality sensing data is essential for decision-making. Traditional sensing methods, however, are often plagued by high deployment costs and incomplete data coverage, significantly limiting their practicality. Despite recent progress, these methods continue to face challenges in maintaining data accuracy, ultimately degrading the Quality of Service (QoS) for IoT applications. To address these limitations, we propose AS-STG, a novel framework that combines an Active Sampling strategy with Spatio-Temporal Graph learning to enable efficient and accurate IoT sensing. At its core, AS-STG is designed to minimize the sampling cost while ensuring the accuracy of the data. The framework begins by analyzing historical data to determine the minimum sampling requirements for accurate inference in subsequent time slots. It then constructs a spatio-temporal graph to model the complex relationships between sensing grids, capturing both spatial and temporal dynamics. To supplement the spatio-temporal information and further optimize representations, we introduce two contrastive learning tasks. Leveraging the refined representation, AS-STG strategically selects information-rich regions for sampling, ensuring that even a sparse subset of samples can provide comprehensive coverage of the entire sensing area. Finally, AS-STG employs matrix completion techniques to reconstruct the complete sensing data from these sparse samples. Extensive experiments on real-world datasets demonstrate that AS-STG significantly outperforms baselines in terms of inference accuracy, cost-efficiency, and scalability. By effectively reducing sampling costs without compromising QoS, AS-STG offers a robust and scalable solution for dynamic IoT sensing systems.**

***Index Terms*—IoT sensing, active sampling, spatio-temporal graph, matrix completion**

I. Introduction

The proliferation of the Internet of Things (IoT) has endowed devices with the ability to "communicate", paving the way for smart applications. By integrating Artificial Intelligence (AI) algorithms for sensing data analytics, these applications can autonomously process and interpret complex data streams, enabling data-driven decision-making. A prominent example is smart transportation systems, where adaptive signal control leverages Vehicle-to-Infrastructure (V2I) data to dynamically optimize traffic flow, significantly alleviating urban congestion and enhancing overall transportation efficiency [1], [2].

Traditional sensing methods rely heavily on the large-scale deployment of IoT devices (such as sensors), which have been widely adopted in various applications [3]. The in-depth analysis of sensing data provides valuable insights for stakeholders across industries [4], [5]. However, achieving full coverage remains challenging due to environmental instability and privacy concerns in certain areas. Coverage is a critical metric for assessing QoS, as it determines whether sufficient data is available to support reliable decision-making [6]–[8]. In large-scale and fine-grained IoT applications, the data collected by IoT devices are often incomplete, random, and sparse, leading to degraded performance [9], [10].

To address these challenges, an ideal approach is to maximize the information of sampled data and infer unsampled data from this information, rather than incurring high costs to collect the entire data. This approach is justified by the strong spatio-temporal correlations inherent in IoT environments, which often exhibit low-rank properties [11], [12]. These properties make cost-effective sensing through Matrix Completion (MC) feasible, where the inherent structure of the data allows for the reconstruction of missing values from observed samples [13]. While some studies have applied MC to the sensing domain, they primarily focus on optimizing MC performance, and do not actively provide strategies for sampling and input location selection [14]–[16]. Given the spatio-temporal correlations, different locations carry varying amounts of information. Therefore, only optimizing MC algorithms may not fully mitigate data inference errors caused by insufficient sampling information.

In this paper, we formulate the sensing process as an active sampling problem based on matrix completion, as illustrated in Fig. 1. Given a sensing area, we divide it into N grids (e.g. $N = 4 \times 4$). At each time slot, k key grids are selected for IoT device placement, obtaining a sampling vector (in our

*Corresponding author: Huiling Qin.

979-8-3315-4940-4/25/$31.00 © 2025 IEEE

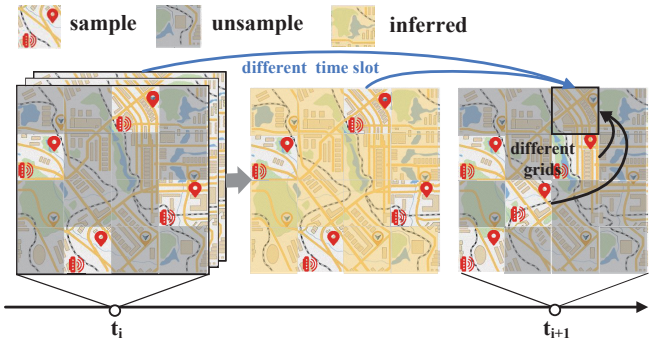


Fig. 1. Illustration of the active sampling problem.

example, $k = 4$, that is 4 IoT devices are placed in a 16-grid setup). By integrating this vector with historical data, matrix completion techniques can effectively estimate the sensing data for unsampled grids. This process focuses on balancing sampling costs and inference accuracy, which are closely tied to the choice of sampling grids. Different grids provide varying amounts of information, directly impacting MC performance. While sampling more grids increases the likelihood of capturing sufficient information, it also raises costs and may introduce redundancy due to spatio-temporal correlations. Conversely, sampling too few grids may lead to insufficient information, severely degrading inference accuracy. Existing researches have shown that for a matrix of size $a \times b$ with rank r, where $r \ll min(a, b)$, accurate inference of unsampled data is possible if the number of sampled grids exceeds a certain threshold [17], [18]. Thus, in our active sampling problem, two challenges should be addressed: i) How to determine the sample size to minimize IoT device deployment costs? ii) How to select sampling grids to maximize the accuracy of matrix completion?

In this paper, we propose AS-STG (Active Sampling strategy with Spatio-Temporal Graph learning) to achieve both cost-effective and high accurate sensing. To address the first challenge, we formulate active sampling as a sparse matrix completion problem. By deriving the necessary conditions to span the complete matrix space, we determine the minimum sample size required for accurate inference, ensuring that the sampling cost is optimized without compromising data quality. For the second challenge, we design a spatio-temporal graph learning model that carefully selects grids to sample in each upcoming time slot. This model leverages historical data to account for the information disparities among different grids and their spatio-temporal correlations. To be further aware of the diverse spatio-temporal patterns among different grids, we introduce two contrastive learning tasks, which is helpful for selecting the optimal sampling grids.

In summary, this paper makes the following contributions:

- We reformulate the IoT sensing process as an active sampling optimization problem, integrating matrix completion to achieve high-accuracy sensing with low sampling cost.
- We propose a spatio-temporal graph learning model that combines grid-based spatial graphs with time series data, effectively capturing the relationships between grids over time. To optimize the model's representational ability, we introduce contrastive learning tasks to supplement the spatio-temporal information.
- Extensive experiments on three real-world datasets demonstrate the consistent superiority of AS-STG across various settings. For example, on the NYCBike dataset, AS-STG achieves approximately a 10% improvement in MAE, about a 16% improvement in MAPE, and around a 7% improvement in RMSE compared to the second-best baseline.

II. Related Works

Leveraging sampled information to infer unsampled data provides a feasible approach to sensing, similar to the sparse mobile crowdsensing problem. This section discusses sparse mobile crowdsensing, followed by its common solution, matrix completion.

A. Sparse Mobile CrowdSensing

Mobile CrowdSensing (MCS) leverages the ubiquity of smart devices carried by mobile users to perform a wide range of urban sensing tasks [19]. However, due to constrained sensing budgets, it is often infeasible to recruit sufficient users for comprehensive data collection. To address this limitation, sparse MCS has emerged as a promising paradigm, which recruits a limited number of mobile users to sample data and infers the state of unsampled areas [20]. This approach has demonstrated significant value in IoT applications. For example, Liu et al. [21] and He et al. [22] develop systems for urban air pollution monitoring and signal mapping using sparse MCS techniques.

Mobile user selection is a critical component of sparse MCS, as it directly impacts the cost-efficiency and data quality of the sensing process. Xie et al. [6] propose an Active Sparse Mobile Crowd Sensing (AS-MCS) framework that leverages bipartite graph modeling to actively select mobile users based on the informativeness of their locations. This method dynamically adjusts mobile user selection in each time slot to ensure that the sampled data provides maximal coverage and minimizes inference errors. Building on this, Zhu et al. [23] introduce a cost-quality beneficial cell selection method consisting of three steps: information modeling, cost estimation, and cost-quality beneficial cell selection. They formulate the mobile user selection as a bi-objective optimization problem, aiming to maximize the informativeness of selected subareas while minimizing total sensing costs. Liu et al. [24] further advance the field by incorporating a quality-aware incentive mechanism that dynamically adjusts mobile user rewards based on the importance of their contributions. They use reinforcement learning to estimate the significance of different regions and selects users to maximize data completeness.

B. Matrix Completion

Matrix completion has proven to be a powerful technique for inferring missing entries in data matrices, particularly when the matrix exhibits low-rank properties [25], [26]. Matrix Factorization (MF) is one of the most widely used tools for matrix completion. The core idea is to decompose the original matrix into the product of simpler matrices, capturing low-rank features and leveraging spatio-temporal correlations to infer missing data. A notable method in this area is Singular Value Thresholding (SVT) [27], which iteratively applies singular value decomposition (SVD) and soft-thresholding to recover low-rank matrices. This method has laid the foundation for many subsequent advancements in matrix completion.

Yu et al. [28] present a Temporal Regularized Matrix Factorization (TRMF) framework, which incorporates a trained time series module to capture temporal correlations, and uses scalable MF to infer data for unobserved time points. Wan et al. [29] integrate deep learning with sparse MCS to enhance matrix completion accuracy, leveraging neural networks to model complex data patterns.

The aforementioned works have improved the accuracy and efficiency of sparse sensing from various angles. However, the inference accuracy still remains poor due to the neglect of hidden spatio-temporal correlations between regions. To address this issue, we propose the AS-STG framework, which better captures and optimizes spatio-temporal correlations by constructing the spatio-temporal graph, designing augmentation components, and employing contrastive learning.

III. Preliminaries

A. Definitions

Definition 1. ***Grid Graph***. *A grid graph is defined as* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$*, where* $\mathcal{V}$ *is the set of grids and* $\mathcal{E}$ *is the set of edges, indicating the connectivity between grids. The sensing area is divided into* N *non-overlapping grids. The set of grids in the sensing area can be denoted by* $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$*. The adjacency matrix of* $\mathcal{G}$ *is denoted as* $\mathcal{A} \in \mathbb{R}^{N \times N}$*, and* $a_{ij} = 1$ *if* $v_i, v_j \in \mathcal{V}$ *and* $\varepsilon_{ij} \in \mathcal{E}$*.*

Definition 2. ***Sensing Matrix***. *The sensing data from* N *grids over a time window* T *can be represented as a* $N \times T$ *sensing matrix* $\boldsymbol{X}^{(T)}$*, where* $\boldsymbol{X}^{(T)} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(T)}\}$*. The data from a grid correspond to a row in matrix* $\boldsymbol{X}^{(T)}$*, while a column in matrix* $\boldsymbol{X}^{(T)}$ *records the data from all grids within a time slot.*

Definition 3. ***Sampling Vector***. *The sampling vector at time slot* $T+1$ *is denoted as* $\boldsymbol{h}^{(T+1)} \in \mathbb{R}^N$*. It is initially set with all elements equal to* 0*. After the sampling at time slot* $T+1$ *is completed, the corresponding positions in* $\boldsymbol{h}^{(T+1)}$ *are updated with the sampled values.*

B. Problem Statement

Given the historical sensing matrix and the grid graph, we aim to infer the entire sensing data at the current time slot with low sampling cost and high inference accuracy.

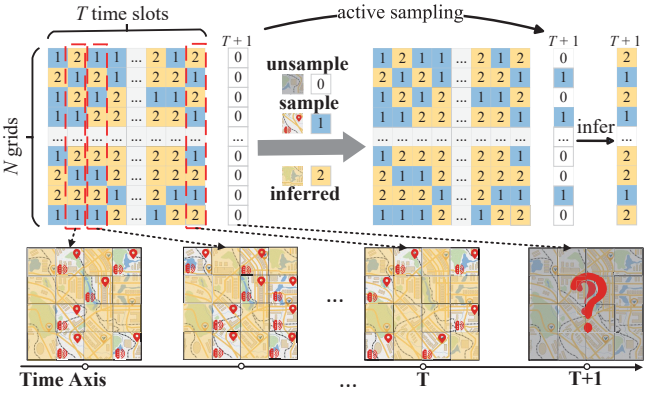


Fig. 2. Description of sensing process.

As illustrated in Fig. 2, for the left sensing matrix $\boldsymbol{X}^{(T)}$ over T time slots, the white grids labeled "0" represent the unsampled data, the blue grids labeled "1" denote the sampled data, and the yellow grids labeled "2" show the data inferred from past time slots. We initially employ this information to determine the sampling grids at time $T+1$, thereby facilitating the inference of unsampled data.

The flag vector $\boldsymbol{z}^{(t)} = [z_1^{(t)}, z_2^{(t)}, \cdots, z_N^{(t)}]^\top$ is used to indicate whether a grid is sampled at time slot t. The sampling status of the i-th grid at the t-th time slot is given by Eq. (1):

$$z_i^{(t)} = \begin{cases} 1 & \text{if sampled;} \\ 0 & \text{others.} \end{cases} \tag{1}$$

$\boldsymbol{x}^{(t)} \odot \boldsymbol{z}^{(t)}$ represents the sampled data at time slot t, and $\boldsymbol{x}^{(t)} \odot (\boldsymbol{1} - \boldsymbol{z}^{(t)})$ is the inferred data for the unsampled grids. We assume a uniform sampling cost of 1 for each grid, such that the total sampling cost at time $T+1$ is given by the L1-norm of the flag vector, i.e., $\|\boldsymbol{z}^{(T+1)}\|_1$.

Formally, the problem can be formulated as an optimization task:

$$\mathcal{F}_1(\boldsymbol{X}^{(T)}, \mathcal{G}) \rightarrow [\theta, \boldsymbol{z}^{(T+1)}, \boldsymbol{x}^{(T+1)}], \tag{2}$$

where $\theta = \|\boldsymbol{z}^{(T+1)}\|_1$.

IV. Methodology

The overview framework of AS-STG is illustrated in Fig. 3, including three main modules: i) the **Sample Size Determination** module is applied to identify the minimum sampling cost required for accurate inference; ii) the **Active Sampling** module is used to select sampling grids within the given cost; and iii) the **Data Inference** module is employed to optimize the inference results.

A. Sample Size Determination

It is straightforward to determine the number of samples needed using the properties of the matrix. For example, consider the matrix $\mathcal{X}$ with $rank(\mathcal{X})$, $rank(\mathcal{X})$ linearly independent vectors can form the basis of the matrix, spanning the entire matrix space. It is evident that the number of samples is related to the rank of the sensing matrix.

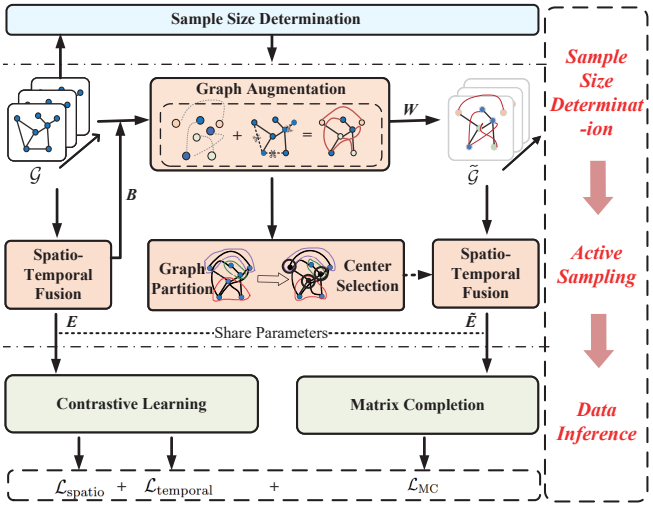


Fig. 3. Overall framework of AS-STG.

In our scenario, given the sensing matrix $\boldsymbol{X}^{(T)}$ with the rank of r, to minimize the sampling cost, we need to know the rank of $\boldsymbol{X}^{(T+1)}$. According to Theorem 1, the rank of $\boldsymbol{X}^{(T+1)}$, denoted as r', satisfies $r' \in [r, r+1]$. This implies that the minimum sampling cost required to accurately reconstruct $\boldsymbol{X}^{(T+1)}$ is at least r+1.

Theorem 1. *Given the matrix $\boldsymbol{X}^{(T)}$ with rank r, the rank of the matrix $\boldsymbol{X}^{(T+1)}$ satisfies:*

$$r' \in [r, r+1]. \tag{3}$$

Proof. The matrix $\boldsymbol{X}^{(T+1)}$ is obtained by adding a column to matrix $\boldsymbol{X}^{(T)}$, which can be expressed as:

$$\boldsymbol{X}^{(T+1)} = [\boldsymbol{X}^{(T)}, \boldsymbol{x}^{(T+1)}]. \tag{4}$$

We need to consider the linear relationship between the new column $\boldsymbol{x}^{(T+1)}$ and the existing columns of $\boldsymbol{X}^{(T)}$. If the new column $\boldsymbol{x}^{(T+1)}$ is a linear combination of the columns of $\boldsymbol{X}^{(T)}$, then $\boldsymbol{x}^{(T+1)}$ does not add any new dimension to the column space. Therefore, the rank remains the same, that is, $r' = r$.

If the new column $\boldsymbol{x}^{(T+1)}$ is linearly independent of the columns of $\boldsymbol{X}^{(T)}$, then $\boldsymbol{X}^{(T)}$ adds a new dimension to the column space, increasing the rank by 1, namely $r' = r + 1$. Thus, Eq. (3) is proved. □

Based on this theorem, we dynamically determine the rank of $\boldsymbol{X}^{(T+1)}$ by analyzing the historical window data. It allows us to adaptively obtain the minimum sampling cost θ required for accurate inference, ensuring both efficiency and robustness in resource-constrained IoT environments.

B. Active Sampling

The active sampling module strategically select sampling grids by leveraging spatio-temporal dependencies in historical data. It consists of three key components: Spatio-Temporal Fusion, Graph Augmentation, and Graph Partitioning & Center Selection.

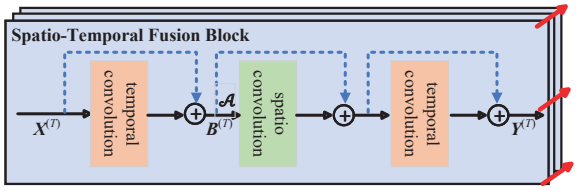


Fig. 4. Spatio-temporal fusion component.

1) Spatio-Temporal Fusion: The spatio-temporal fusion component learns complex spatio-temporal correlations through a series of spatio-temporal fusion blocks. Inspired by [30], our spatio-temporal fusion block employs a "sandwich structure". As shown in Fig. 4, by alternating between Temporal Convolutions (TC) and Spatial Convolutions (SC), it simultaneously captures and integrates spatio-temporal features, thereby enhancing the module's representation and generalization capabilities. This process can be succinctly described as follows:

$$\mathrm{TC}(\boldsymbol{X}^{(T)}) \longrightarrow \boldsymbol{B}^{(T)}, \tag{5}$$

$$\mathrm{TC}(\mathrm{SC}(\boldsymbol{B}^{(T)}, \mathcal{A})) \longrightarrow \boldsymbol{Y}^{(T)}, \tag{6}$$

where $\boldsymbol{B}^{(T)} \in \mathbb{R}^{N \times D \times T'}$ and $\boldsymbol{Y}^{(T)} \in \mathbb{R}^{N \times D \times T''}$ denote the embedding of the spatio-temporal graph at different stages, respectively. D is the embedding dimensionality, and T' and T'' are the temporal dimensionality after convolutions. After stacking multiple blocks, we obtain the final embedding $\boldsymbol{E}^{(T)} \in \mathbb{R}^{N \times D}$ for grids in the graph, encompassing the temporal dimensionality.

2) Graph Augmentation: Traditional approaches predominantly rely on pre-defined graphs for modeling spatial relationships between grids. While these static graphs provide a basic structure, they often fail to capture latent similarities and intrinsic relationships between grids, thus limiting the effectiveness of the resulting graph representations [31]. To address this limitation, we propose the graph augmentation component.

For grid v_i, its temporal embedding $\boldsymbol{s}_i$ across several time steps can be represented as:

$$\boldsymbol{s}_i = \mathrm{TC}^{(m)}(\boldsymbol{b}_i), \tag{7}$$

where m represents multiple temporal convolutions, aggregating information across all time steps and $\boldsymbol{b}_i \in \mathbb{R}^{D \times T'}$ denotes the embedding of grid v_i in $\boldsymbol{B}^{(T)}$.

We use the cosine distance of the temporal embeddings to measure the similarity between two grids. For example, the similarity between grid v_i and grid v_j can be represented as:

$$w_{ij} = \frac{\boldsymbol{s}_i^\top \boldsymbol{s}_j}{\|\boldsymbol{s}_i\| \|\boldsymbol{s}_j\|}. \tag{8}$$

Elements $w_{ij} (i \in [1, N], j \in [1, N])$ construct the similarity matrix $\boldsymbol{W} \in \mathbb{R}^{N \times N}$. The matrix $\boldsymbol{W}$ is symmetric with all

Algorithm 1: The adaptive random graph augmentation algorithm.

Input: the similarity matrix $\boldsymbol{W}$ the grid graph $\mathcal{G}$, $\mathcal{A}$
Output: the augmented graph $\tilde{\mathcal{G}}, \tilde{\mathcal{A}}$

```
for i = 1; i ≤ N; i++ do
    for j = i+1; j ≤ N; j++ do
        according to Eq. (10), compute ρ_ij;
        Generate a random number α ∈ [0, 1];
        if a_ij == 1 then
            p_ij = ρ_ij;
            if α ≤ p_ij then
                a_ij = 0, remove ε_ij from E;
            else
                continue;
        else
            p_ij = 1 − ρ_ij;
            if α ≤ p_ij then
                a_ij = 1, add ε_ij into E;
            else
                continue;
After updating A and E, Ã and Ẽ are obtained;
return G̃ = {V, Ẽ}, Ã.
```

diagonal elements set to 1. A higher w_{ij} indicates that the data patterns between the grids are more similar and vice versa.

The similarity matrix $\boldsymbol{W}$ allows us to re-evaluate the adjacency relations between grids. Inspired by [32], we design an adaptive random graph augmentation algorithm based on the similarity matrix. Specifically, the addition or removal of the edge ε_{ij} is performed based on a probability p_{ij}, which is drawn from a Bernoulli distribution. The probability p_{ij} can be described as follows:

$$p_{ij} = \begin{cases} \rho_{ij} & a_{ij} = 1 \\ 1 - \rho_{ij} & a_{ij} = 0, \end{cases} \tag{9}$$

$$\rho_{ij} = \frac{\max(\boldsymbol{W}\dot{(}\boldsymbol{1} - \boldsymbol{I})) - w_{ij}}{\max(\boldsymbol{W}\dot{(}\boldsymbol{1} - \boldsymbol{I})) - \overline{\boldsymbol{W}\dot{(}\boldsymbol{1} - \boldsymbol{I})}}, \tag{10}$$

where $\max(\boldsymbol{W}\dot{(}\boldsymbol{1} - \boldsymbol{I}))$ represents the maximum value of the off-diagonal elements in the matrix $\boldsymbol{W}$, and $\overline{\boldsymbol{W}\dot{(}\boldsymbol{1} - \boldsymbol{I})}$ denotes their average value. If grids v_i and v_j are adjacent, we remove the edge ε_{ij} with probability ρ_{ij} and set $a_{ij} = 0$. ρ_{ij} decreases as the similarity w_{ij} increases. If grids v_i and v_j are not adjacent, an edge is added with a probability of $1 - \rho_{ij}$, meaning that $a_{ij} = 1$. The probability of adding the edge follows the trend of the similarity w_{ij}. After the above augmentation process, an augmented graph $\tilde{\mathcal{G}} = (\mathcal{V}, \tilde{\mathcal{E}})$ can be obtained. The adjacency matrix $\mathcal{A}$ is also augmented to $\tilde{\mathcal{A}}$. In order to make it more easy to understand, we give a pseudo code flow table (Alg. 1).

3) Graph Partitioning & Center Selection: The graph augmentation component has generated an augmented graph, where grids with similar data patterns become closer to each other, facilitating their grouping into the same cluster.

Spectral clustering provides an effective approach for partitioning such graphs, primarily based on the Laplacian matrix $\boldsymbol{L}$. In our setting, $\boldsymbol{L}$ is derived from the similarity matrix $\boldsymbol{W}$, and its representation is as follows:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}, \tag{11}$$

$$\boldsymbol{D} = \begin{bmatrix} \sum_{i \neq 1} w_{1i} & 0 & \cdots & 0 \\ 0 & \sum_{i \neq 2} w_{2i} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sum_{i \neq N} w_{Ni} \end{bmatrix}. \tag{12}$$

After obtaining the Laplacian matrix $\boldsymbol{L}$, the first step is to perform eigenvalue decomposition to derive a new matrix composed of its eigenvectors. This new matrix is then used in the K-Means algorithm to partition the graph into $r+1$ clusters. Compared to other methods, spectral clustering offers an effective clustering solution by reducing computational complexity and better adapting to the data distribution characteristics.

For the clustered groups, several simple and effective methods can be used to select sampling grids (i.e., cluster centers), such as degree centrality, eigenvector centrality, and PageRank. In this paper, we primarily use degree centrality for this purpose. Degree centrality is one of the simplest centrality measures, evaluating the importance of a node by calculating its degree (the number of edges directly connected to it). In our scenario, the degree of a grid node directly reflects the number of grids with similar data patterns.

We use $\mathcal{F}_2$ to represent the centrality selection method, and the process of obtaining sampling grids can be expressed as:

$$\begin{aligned} \nu_i^{(T+1)} &= \underset{\nu_i \in \mathcal{V}_i}{\arg\max}\, \mathcal{F}_2(\tilde{\mathcal{A}}_i, \tilde{\mathcal{V}}_i), 1 \leq i \leq r+1, \\ \nu^{(T+1)} &= \{\nu_1^{(T+1)}, \nu_2^{(T+1)}, \cdots, \nu_{r+1}^{(T+1)}\}, \end{aligned} \tag{13}$$

where $\tilde{\mathcal{V}}_i$ and $\tilde{\mathcal{A}}_i$ represent the clusters of grids obtained after partitioning and the corresponding augmented adjacency matrix, respectively. Based on $\nu^{(T+1)}$, we obtain the sampling flag vector $\boldsymbol{z}^{(T+1)}$ and sampling vector $\boldsymbol{h}^{(T+1)}$.

C. Data Inference

After completing the previous modules, the sensing matrix $[\boldsymbol{X}^{(T)}, \boldsymbol{h}^{(T+1)}]$ could be obtained. We employ matrix factorization to infer the missing data at unsampled grids, which correspond to the locations indicated by $\boldsymbol{1} - \boldsymbol{z}^{(T+1)}$. Specifically, the sensing matrix $[\boldsymbol{X}^{(T)}, \boldsymbol{h}^{(T+1)}]$ is factorized into the product of two low-rank factor matrices: $\boldsymbol{U}^{(T+1)} \in \mathbb{R}^{N \times (r+1)}$ and $\boldsymbol{V}^{(T+1)} \in \mathbb{R}^{(T+1) \times (r+1)}$. The goal of MF is to minimize the error ϵ between $[\boldsymbol{X}^{(T)}, \boldsymbol{h}^{(T+1)}]$ and $\boldsymbol{U}^{(T+1)} \times \boldsymbol{V}^{(T+1)\top} \odot [\boldsymbol{1}, \boldsymbol{1}, \cdots, \boldsymbol{1}, \boldsymbol{z}^{(T+1)}]$, ensuring that the missing values in the sensing matrix are accurately inferred. The missing values at unsampled grids could be computed as $\hat{\boldsymbol{x}}^{(T+1)} \odot (\boldsymbol{1} - \boldsymbol{z}^{(T+1)})$, where $\hat{\boldsymbol{x}}^{(T+1)}$ represents the inferred results at time $T+1$. The complete inference process is illustrated in Fig. 5.

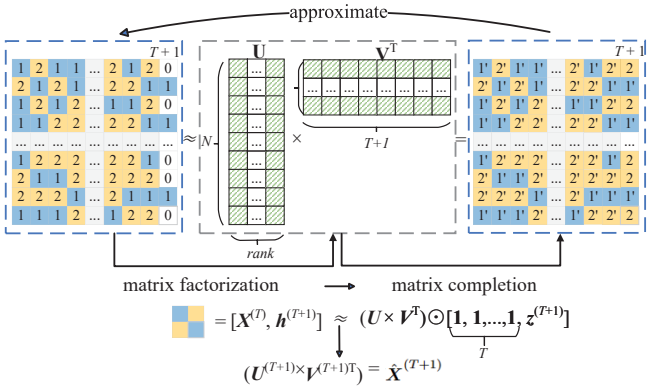


Fig. 5. The data inference process based on matrix factorization.

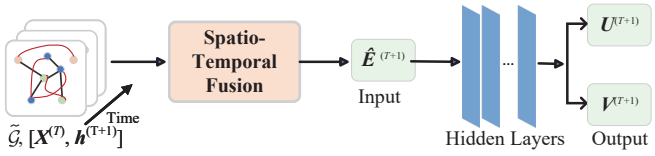


Fig. 6. The process of obtaining factor matrices.

To ensure high-quality initialization of the factor matrices, which is critical for the convergence of the matrix completion process, we leverage spatio-temporal graph embeddings for training. These embeddings effectively capture and reflect the intrinsic characteristics of grids, enabling the derivation of factor matrices that are both representative and well-suited for subsequent optimization. The process is illustrated in Fig. 6 and consists of the following steps. Firstly, the grid graph $\tilde{\mathcal{G}}$ and the concatenated data $[\boldsymbol{X}^{(T)}, \boldsymbol{h}^{(T+1)}]$ are fed into spatio-temporal fusion component to generate the embeddding $\hat{\boldsymbol{E}}^{(T+1)}$. Next, a Multi-Layer Perceptron (MLP) is trained to derive the factor matrices $\boldsymbol{U}^{(T+1)}$ and $\boldsymbol{V}^{(T+1)}$ from $\hat{\boldsymbol{E}}^{(T+1)}$. Finally, the complete sensing data is reconstructed through simple operations on the factor matrices.

D. Model Loss

In the AS-STG learning process, the selected $r+1$ key grids are used to generate the embedding $\hat{\boldsymbol{E}}^{(T+1)}$, which serves as the foundation for deriving two spatio-temporally related factor matrices, $\boldsymbol{U}^{(T+1)}$ and $\boldsymbol{V}^{(T+1)}$. The quality of these factor matrices is optimized through a matrix completion loss, defined as:

$$\mathcal{L}_{\mathrm{MC}} = \|[\boldsymbol{X}^{(T)}, \boldsymbol{h}^{(T+1)}] - (\boldsymbol{U}^{(T+1)} \times \boldsymbol{V}^{(T+1)\top}) \odot \boldsymbol{Z}^{(T+1)}\|_F^2, \tag{14}$$

where $\boldsymbol{Z}^{(T+1)}$ is a simple expression of $[\mathbf{1}, \mathbf{1}, \cdots, \mathbf{1}, \boldsymbol{z}^{(T+1)}]$. This loss ensures that the factor matrices accurately reconstruct the sensing data while preserving the underlying spatio-temporal structure.

Besides to optimizing the factor matrices, we also focus on improving the quality of the sptiao-temporal embedding, which significantly affects the selection of representative grids. To enhance the robustness and generalizability of the embeddings, we introduce contrastive learning tasks applied separately in the spatial and temporal dimensions. These tasks encourage the model to develop discrimination ability by explicitly capturing both spatial and temporal heterogeneity.

In the spatial learning dimension, we design a cluster-based contrastive learning task that maps grids into multiple potential representation clusters. Specially, we map the N grids into $r+1$ clusters based on the embedding $\boldsymbol{E}^{(T)}$ and $\tilde{\boldsymbol{E}}^{(T)}$ from the original and augmented graphs, respectively. It yields the probability of each grid being assigned to each cluster. By maximizing the clustering similarity between $\boldsymbol{E}^{(T)}$ and $\tilde{\boldsymbol{E}}^{(T)}$, we can learn features that remain invariant to different transformation operations, thereby making the embeddings more compact and distinctive. According to [32] and [33], the learning loss function can be defined as:

$$\mathcal{L}_{\mathrm{spatio}} = l(\boldsymbol{E}^{(T)}, \tilde{\boldsymbol{E}}^{(T)}) + l(\tilde{\boldsymbol{E}}^{(T)}, \boldsymbol{E}^{(T)}), \tag{15}$$

where $l(\boldsymbol{E}, \tilde{\boldsymbol{E}})$ represents the loss of $\tilde{\mathcal{G}}$ relative to $\mathcal{G}$, and $l(\tilde{\boldsymbol{E}}, \boldsymbol{E})$ has a symmetric meaning (to simplify the mathematical expression, the time index (T) has been omitted here).

$$l(\boldsymbol{E}, \tilde{\boldsymbol{E}}) = -\frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{r+1}\mathcal{S}((\boldsymbol{c}_k \cdot \boldsymbol{e}_j)/\tau)_k \log \mathcal{S}((\tilde{\boldsymbol{c}}_k \cdot \tilde{\boldsymbol{e}}_j)/\epsilon)_k, \tag{16}$$

where $\mathcal{S}((\boldsymbol{c}_k \cdot \boldsymbol{e}_j)/\tau)_k$ and $\mathcal{S}((\tilde{\boldsymbol{c}}_k \cdot \tilde{\boldsymbol{e}}_j)/\epsilon)_k$ represent the assignment probability of grid v_j to cluster k in graph $\mathcal{G}$ and augmented graph $\tilde{\mathcal{G}}$, respectively. τ is the temperature parameter that controls the smoothness of the Softmax output, $\boldsymbol{c}_k \in \mathbb{R}^D$ is the cluster embedding and $\boldsymbol{e}_j$ is the grid embedding of grid v_j in graph $\mathcal{G}$. ϵ, $\tilde{\boldsymbol{c}}$ and $\tilde{\boldsymbol{e}}_j$ are defined similarly to the above. $\mathcal{S}(\cdot)$ is the Softmax function, and the full expression of $\mathcal{S}((\boldsymbol{c}_k \cdot \boldsymbol{e}_j)/\tau)_k$ is:

$$\mathcal{S}((\boldsymbol{c}_k \cdot \boldsymbol{e}_j)/\tau)_k = \frac{\exp((\boldsymbol{c}_k \cdot \boldsymbol{e}_j)/\tau)}{\sum_{m=1}^{r+1}\exp((\boldsymbol{c}_m \cdot \boldsymbol{e}_j)/\tau)}. \tag{17}$$

In the temporal learning dimension, we design a contrastive learning task to capture the differences between time slots. For each grid, at each time slot, its relationship with other grids at the same time slot is considered a positive pair, while its relationship with other time slots is considered a negative pair. Positive pairs help capture consistent trends within the same time slot, while negative pairs help capture temporal heterogeneity across different time slots. In this process, the loss function can be defined as the follow:

$$\mathcal{L}_{\mathrm{temporal}} = -\frac{1}{N \cdot T}\sum_{t=1}^{T}\sum_{i=1}^{N}\left(\sum_{j\neq i}\log(\pi_{v_i,v_j,t}) + \sum_{t'\neq t}\log(1-\pi_{v_i,t,t'})\right), \tag{18}$$

where $\pi_{v_i,v_j,t}$ is the similarity score between grid v_i and grid v_j at time slot t and $\pi_{v_i,t,t'}$ is the similarity score between grid v_i at time step t and time step t'. They can be obtained through the dot product of the grid embeddings, mapped to the range [0, 1] via the sigmoid function.

Based on the above steps, we can obtain a comprehensive optimization objective:

$$\mathcal{L}_{\text{syn}} = \lambda_1 \mathcal{L}_{\text{MC}} + \lambda_2 \mathcal{L}_{\text{spatio}} + \lambda_3 \mathcal{L}_{\text{temporal}}, \tag{19}$$

where λ_1,λ_2, and λ_3 represent loss weights.

E. Model Training

The overall training process of our AS-STG can be described as follows. i) Given the spatio-temporal graph $\mathcal{G}$ and $\boldsymbol{X}^{(T)}$, we first determine the optimal number of samples based on the rank of the matrix. The spatio-temporal fusion component then generates the initial embedding $\boldsymbol{E}^{(T)}$. ii) We preform the graph augmentation component to refine $\mathcal{G}$ to $\tilde{\mathcal{G}}$, which can then be used to obtain the embedding $\tilde{\boldsymbol{E}}^{(T)}$. iii) Using $\tilde{\mathcal{G}}$ and the estimated sample size $r+1$, the graph partitioning & center selection component is performed to select the sampling grids and get sampling vector $\boldsymbol{h}^{(T+1)}$. iv) The augmented graph $\tilde{\mathcal{G}}$ and $[\boldsymbol{X}^{(T)}, \boldsymbol{h}^{(T+1)}]$ are fed into the data inference module and derives the factor matrices $\boldsymbol{U}^{(T+1)}$ and $\boldsymbol{V}^{(T+1)}$. v) Compute the loss $\mathcal{L}_{\text{syn}}$ using $\boldsymbol{X}^{(T)}$, $\boldsymbol{h}^{(T+1)}$, $\boldsymbol{z}^{(T+1)}$, $\boldsymbol{U}^{(T+1)}$, $\boldsymbol{V}^{(T+1)}$, $\boldsymbol{E}^{(T)}$ and $\tilde{\boldsymbol{E}}^{(T)}$. vi) Train until convergence. vii) Update $\boldsymbol{U}^{(T+1)}$ and $\boldsymbol{V}^{(T+1)}$ to get the sensing data $\boldsymbol{x}^{(T+1)}$.

V. Experiments

In this section, we perform a series of experiments designed to evaluate our AS-STG's performance by answering the following key questions:

- **RQ1:** How accurate is our proposed method in comparison to baselines?
- **RQ2:** How effective are the individual components we designed in enhancing the overall performance?
- **RQ3:** How beneficial are the selected sampling grids for facilitating accurate matrix completion?
- **RQ4:** How effective is our method in reducing sampling costs with maintaining inference accuracy?

A. Experimental Settings

1) Datasets: To evaluate the performance of AS-STG, we conduct experiments on three popular real-world datasets. Tab. I provides more detailed information on these datasets.

TABLE I
Description and Statistics of Datasets

Dataset	**NYCBike** [34]	**NYCTaxi** [35]	**U-Air** [36]
Data Type	Bike Rent	Taxi GPS	PM2.5
Location	New York	New York	Beijing, etc
Grid	128	200	437
Time Interval	1 hour	30 min	1 hour
Time Span	04/01/2014-09/30/2014	01/01/2015-03/01/2015	05/01/2014-04/30/2015

2) Baselines: To comprehensively evaluate the effectiveness of our proposed AS-STG, we compare it with the following methods, which include three categories: **matrix completion algorithms**, **sampling grid selection methods**, and **spatio-temporal modeling methods**. Matrix completion algorithms are used to recover missing parts of the sensing data, sampling grid selection algorithms optimize the data sampling process, and spatio-temporal modeling methods capture the spatio-temporal features in the data to predict future data.

(1) **Matrix Completion Algorithms**:

• **SVT** [27]: Singular Value Thresholding, which approximates the minimum nuclear norm matrix through iterative singular value soft-thresholding, generating a new matrix at each step and gradually approaching the optimal solution.

• **DMF** [29]: Deep matrix factorization, which integrates deep learning to extract superior features from the raw data space, thereby improving the initialization accuracy of matrix factorization.

• **TRMF** [28]: Temporal Regularized Matrix Factorization, which infers missing data by capturing the latent relationships between time points in a time series.

(2) **Sampling Grid Selection Methods**:

• **OS-MCS** [24]: Online Sparse Mobile Crowd-Sensing, which estimates the importance of spatio-temporal regions using a reinforcement learning model, selects important grids for data collection and performs matrix completion.

• **AS-MCS** [6]: Active Sparse Mobile Crowd-Sensing, which introduces a bipartite graph to represent the environment matrix, and selects grid samples by capturing information differences across different grids.

(3) **Spatio-Temporal Modeling Methods**:

• **AGCRN** [37]: Adaptive Graph Convolutional Recurrent Network, which is designed to automatically capture fine-grained spatial and temporal correlations in correlated time series data using node-specific pattern learning and data-driven graph generation.

• **PDFormer** [38]: Propagation Delay-aware dynamic long-range transFormer, which includes a spatial self-attention module and a delayed perception module to capture spatial dependencies and model time delays.

3) Metrics: In the following experiments, Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) are selected to measure the performance of our model.

4) Parameter Settings: For fairness, we partition the data into training, validation, and test sets in the same manner as the baselines, using a ratio of 7:1:2. The convolution kernels for both spatial and temporal convolutions are set to 3, and the embedding dimension is set to 64. All experiments are conducted on a Linux server (CPU: Intel(R) Xeon(R) Gold 5320 CPU @ 2.20GHz, GPU: NVIDIA A100). We use the ADAM optimizer to train our model, with the training epochs set to 100, the learning rate set to 0.001, and the batch size set to 32.

As analyzed earlier, the number of samples should be set to $r+1$, where r is the rank of the sensing matrix. In general

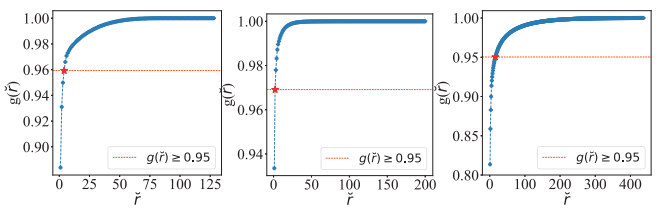
TABLE II
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON THREE DATASETS.

Dataset	Methods	MAE	MAPE(%)	RMSE
NYCBike	SVT	9.04	30.41	15.99
	DMF	6.62	23.77	11.81
	TRMF	6.85	24.26	11.94
	OS-MCS	6.61	26.23	9.61
	AS-MCS	7.29	28.71	10.44
	AGCRN	5.81	26.51	-
	PDFormer	5.36	27.28	7.58
	AS-STG	**4.82**	**22.95**	**7.03**
NYCTaxi	SVT	29.88	29.37	57.83
	DMF	25.72	36.50	46.96
	TRMF	29.69	29.80	56.30
	OS-MCS	27.09	27.99	52.71
	AS-MCS	27.89	18.78	52.75
	AGCRN	13.69	24.01	-
	PDFormer	12.44	21.51	21.90
	AS-STG	**11.62**	**16.92**	**20.71**
U-Air	SVT	26.56	40.35	40.94
	DMF	17.69	32.82	32.66
	TRMF	15.34	24.11	27.69
	OS-MCS	12.05	21.36	21.96
	AS-MCS	15.29	26.78	28.24
	AGCRN	10.53	22.03	22.49
	PDFormer	10.33	17.92	**20.97**
	AS-STG	**10.09**	**17.74**	21.57

definitions, this rank refers to the "precise rank". However, in practical environments, using the precise rank is unrealistic, as any small perturbation in the matrix elements can change the rank [39], [40]. According to Principal Component Analysis (PCA), if a matrix is low-rank, its top $\breve{r}$ singular values occupy most of the total variance. Therefore, we set the number of samples based on the proportion $g(\breve{r})$:

$$g(\breve{r}) = \frac{\sum_{i=1}^{\breve{r}} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2}, \tag{20}$$

where σ_i stands for the singular value. In experiments, to balance completion accuracy and sampling cost, the value of $\breve{r}$ satisfies $\arg\min_{\breve{r}}\{\breve{r} \mid f(\breve{r}) \geq 0.95\}$. Fig. 7 shows the proportion of the top $\breve{r}$ singular values. We set the values of $\breve{r}$ as follows: $\breve{r} = 4$ (NYCBike), $\breve{r} = 2$ (NYCTaxi) and $\breve{r} = 16$ (U-Air). Therefore, in the following experiments, the number of sampling grids for the three datasets are set to: 5 (NYCBike), 3(NYCTaxi) and 17 (U-Air).



(a) NYCBike (b) NYCTaxi (c) U-Air

Fig. 7. The proportion of the total variance occupied by the top $\breve{r}$ singular values.

B. *Accuracy Performance* (**RQ1**)

Tab. II shows the performance comparison of AS-STG against seven baseline methods across three datasets. Our method consistently achieves the best performance across most metrics on all datasets. For example, on the NYCBike dataset, AS-STG attains an MAE of 4.82, MAPE of 22.95%, and RMSE of 7.03, outperforming all competing methods. Compared to PDFformer, the second-best method, AS-STG reduces MAE by approximately 10%, MAPE by 16%, and RMSE by 7%. Similar trends are observed on the NYCTaxi dataset, where AS-STG maintains superior accuracy and efficiency. On the U-Air dataset, although AS-STG's RMSE is slightly higher than PDFformer's (21.57 vs. 20.97), it achieves lower MAE (10.09 vs. 10.33) and MAPE (17.74 vs. 17.92), demonstrating more accurate inference overall.

These improvements can be attributed to AS-STG's enhanced ability to capture complex spatio-temporal correlations. Based on this, grids rich in information are obtained. The spatio-temporal fusion component and graph augmentation techniques enable a deeper understanding of data patterns by integrating spatial topology and temporal dynamics. Furthermore, the incorporation of contrastive learning during training significantly optimizes the spatio-temporal embeddings, leading to more robust and accurate data inference.

C. *Ablation Experiments* (**RQ2**)

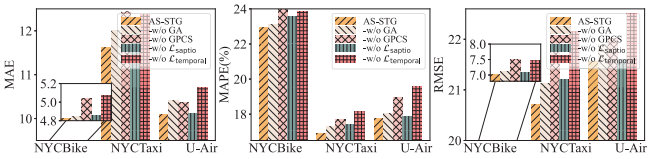
To validate the effectiveness of the individual components in AS-STG, we conduct ablation studies by creating the following variants:

- **-w/o GA**: it removes the graph augmentation component, using the original graph throughout the training process.
- **-w/o GPCS**: it removes the graph partitioning & center selection component, opting for a random method to select the centers.
- **-w/o** $\mathcal{L}_{\text{spatio}}$: it removes $\mathcal{L}_{\text{spatio}}$ during the joint training process.
- **-w/o** $\mathcal{L}_{\text{temporal}}$: it removes $\mathcal{L}_{\text{temporal}}$ during the joint training process.

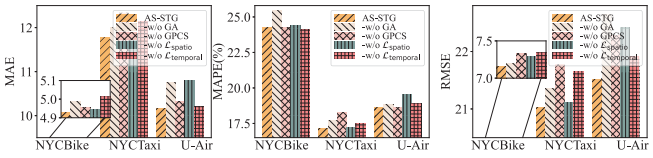
The ablation results, shown in Fig. 8, measure the inference errors for both all grids and unsampled grids. AS-STG consistently outperforms all variants, demonstrating the critical role of each component in the framework. Notably, the importance of individual components varies across datasets. For instance, in the NYCBike dataset, the graph partitioning & center selection component and temporal contrastive learning tasks play particularly significant roles. In contrast, in the U-Air dataset, the graph augmentation component also proves to be highly impactful. These findings highlight the adaptability of AS-STG to different datasets and application scenarios. By optimizing the configuration and parameter settings, AS-STG can achieve stronger adaptability and robustness, making it suitable for a wide range of IoT applications.

D. *Effectiveness of Sampling Grids* (**RQ3**)

To validate the effectiveness of the selected sampling grids in improving matrix completion, we conduct comparative

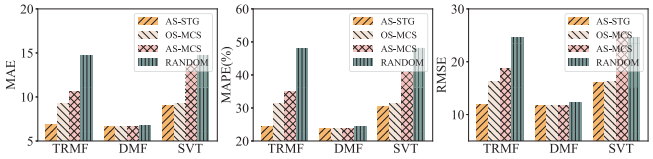


(a) All Grids

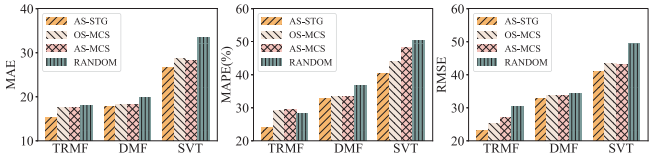


(b) Unsampled Grids

Fig. 8. Ablation Experiments. Fig. 8 (a) shows the inference error for all grids at time T+1 and Fig. 8 (b) shows the inference error for unsampled grids.



(a) NYCBike



(b) U-Air

Fig. 9. Grids Sensitivity Analysis. Validation of the impact and effectiveness of sampling grids in different data inference methods.

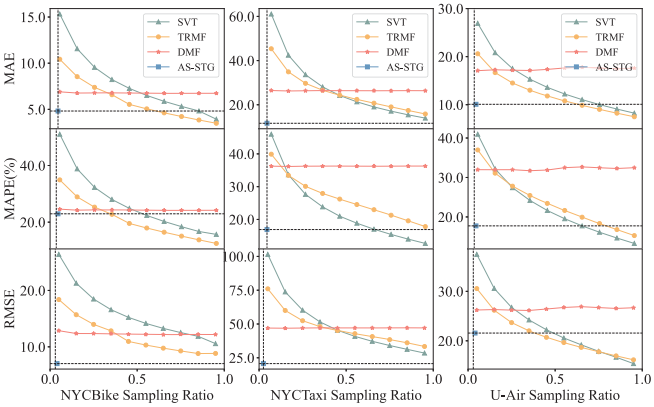


Fig. 10. The impact of sampling ratio on results.

experiments on the NYCTaxi and U-Air datasets. In these experiments, we employ three matrix completion algorithms: SVT, TRMF, and DMF, as well as three different sampling grid selection strategies: OS-MCS, AS-MCS, and random sampling.

As shown in Fig. 9, our proposed sampling method consistently outperforms other strategies across all matrix completion algorithms, demonstrating its robustness and effectiveness in inferring unsampled data. While the impact of grid selection strategies is less pronounced in DMF due to its deep learning-based inference capabilities, our method still achieves the lowest inference error among all strategies. This significant reduction in error underscores the effectiveness of AS-STG in enhancing matrix completion by strategically selecting information-rich grids, thereby improving overall sensing performance.

E. Sampling Costs (**RQ4**)

Fig.10 shows the performance of different matrix completion algorithms under various sampling ratios (5% to 95%). Unlike other algorithms that rely on random sampling with variable ratios, our AS-STG operates with a fixed number of samples, resulting in a consistent and low sampling ratio.

Specifically, AS-STG achieves high inference accuracy at sampling ratios of only 4% for NYCBike, 2% for NYCTaxi, and 4% for U-Air. In contrast, other algorithms require significantly higher sampling ratios (between 35% and 95%) to achieve comparable accuracy. This not only demonstrates the effectiveness of AS-STG in reliably inferring essential data patterns but also highlights its ability to significantly reduce sampling costs while maintaining high accuracy.

VI. Conclusion and Future Work

In this paper, we propose AS-STG, a novel framework for accurate and efficient IoT sensing. By thoroughly analyzing historical data, AS-STG determines the minimum sampling requirements necessary for accurate inference while ensuring robust QoS guarantees. Our framework integrates spatio-temporal fusion, graph augmentation, and contrastive learning to enhance the learning capacity and representational accuracy of spatio-temporal graphs. Leveraging these refined representations, AS-STG strategically identifies and selects information-rich grids for sampling, significantly reducing sampling costs without compromising inference accuracy. Extensive experiments on real-world datasets demonstrate that AS-STG outperforms baselines across various metrics, even at sampling ratios as low as 5%. In future work, we plan to extend AS-STG to handle more complex and dynamic environments. For instance, we will incorporate dynamically changing sampling costs and integrate mobile edge computing to achieve high accurate sensing under resource-constrained conditions. These enhancements will further improve the flexibility and scalability of the sampling process, ensuring accurate sensing results and robust QoS even in challenging scenarios.

Acknowledgment

The above work was supported in part by grants from the National Natural Science Foundation of China (NSFC) (62172046, 62372047), the Beijing Natural Science Foundation (No. 4232028), the Natural Science Foundation of Guangdong Province (2024A1515011323), the Joint Project of Pro-

duction, Teaching and Research of Zhuhai (2220004002686, 2320004002812), the Supplemental Funds for Major Scientific Research Projects of Beijing Normal University, Zhuhai (ZHPT2023002), the Fundamental Research Funds for the Central Universities, Ministry of Education Industry-University Cooperation Collaborative Education Project under 240904497110437, Higher Education Research Topics of Guangdong Association of Higher Education in the 14th Five-Year Plan under 24GYB207, Beijing Normal University Education Reform Project under JX2024139, Key Laboratory of Equipment Data Security and Guarantee Technology, Ministry of Education under Grant No. GDZB2024020100.

REFERENCES

[1] J. Zhang and D. Tao, "Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7789–7817, 2021.

[2] Y. Mei, W. Wang, Y. Liang, Q. Liu, S. Chen, and T. Wang, "Privacy-enhanced cooperative storage scheme for contact-free sensory data in aiot with efficient synchronization," *ACM Transactions on Sensor Networks*, vol. 20, no. 4, pp. 1–19, 2024.

[3] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Data collection and wireless communication in internet of things (iot) using economic analysis and pricing models: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2546–2590, 2016.

[4] O. Ghdiri, W. Jaafar, S. Alfattani, J. B. Abderrazak, and H. Yanikomeroglu, "Offline and online uav-enabled data collection in time-constrained iot networks," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 4, pp. 1918–1933, 2021.

[5] A. Capponi, C. Fiandrino, D. Kliazovich, P. Bouvry, and S. Giordano, "A cost-effective distributed framework for data collection in cloud-based mobile crowd sensing architectures," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 1, pp. 3–16, 2017.

[6] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, and D. Zhang, "Active sparse mobile crowd sensing based on matrix completion," in *SIGMOD 2019*. ACM, 2019, pp. 195–210.

[7] D. Zhang, H. Xiong, L. Wang, and G. Chen, "Crowdrecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *UBICOMP 2014*. ACM, 2014, p. 703–714.

[8] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, "icrowd: Near-optimal task allocation for piggyback crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2010–2022, 2016.

[9] X. Xia, Y. Zhou, J. Li, and R. Yu, "Quality-aware sparse data collection in mec-enhanced mobile crowdsensing systems," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1051–1062, 2019.

[10] Y. Liang, M. Yin, W. Wang, Q. Liu, L. Wang, X. Zheng, and T. Wang, "Collaborative edge server placement for maximizing qos with distributed data cleaning," *IEEE Transactions on Services Computing*, pp. 1–15, 2025.

[11] F. Dai, P. Huang, X. Xu, L. Qi, and M. R. Khosravi, "Spatio-temporal deep learning framework for traffic speed forecasting in iot," *IEEE Internet of Things Magazine*, vol. 3, no. 4, pp. 66–69, 2020.

[12] S. Wang, J. Cao, and P. S. Yu, "Deep learning for spatio-temporal data mining: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3681–3700, 2022.

[13] C. Li, Z. Li, S. Long, P. Qiao, Y. Yuan, and G. Wang, "Robust data inference and cost-effective cell selection for sparse mobile crowdsensing," *IEEE/ACM Transactions on Networking*, pp. 1–16, 2024.

[14] K. Xie, J. Tian, G. Xie, G. Zhang, and D. Zhang, "Low cost sparse network monitoring based on block matrix completion," in *INFOCOM 2021*. IEEE, 2021, pp. 1–10.

[15] E. Wang, M. Zhang, B. Yang, Y. Yang, and J. Wu, "Large-scale spatiotemporal fracture data completion in sparse crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 23, no. 7, pp. 7585–7601, 2024.

[16] E. Wang, M. Zhang, Y. Yang, Y. Xu, and J. Wu, "Exploiting outlier value effects in sparse urban crowdsensing," in *IWQoS 2021*. IEEE, 2021, pp. 1–10.

[17] B. Recht, "A simpler approach to matrix completion," *Journal of Machine Learning Research*, vol. 12, no. 12, pp. 3413–3430, 2011.

[18] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2980–2998, 2010.

[19] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[20] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'hamed, "Sparse mobile crowdsensing: challenges and opportunities," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 161–167, 2016.

[21] T. Liu, Y. Zhu, Y. Yang, and F. Ye, "alc^2: When active learning meets compressive crowdsensing for urban air pollution monitoring," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9427–9438, 2019.

[22] S. He and K. G. Shin, "Steering crowdsourced signal map construction via bayesian compressive sensing," in *INFOCOM 2018*. IEEE, 2018, pp. 1016–1024.

[23] Z. Zhu, B. Chen, W. Liu, Y. Zhao, Z. Liu, and Z. Zhao, "A cost-quality beneficial cell selection approach for sparse mobile crowdsensing with diverse sensing costs," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3831–3850, 2020.

[24] W. Liu, E. Wang, Y. Yang, and J. Wu, "Worker selection towards data completion for online sparse crowdsensing," in *INFOCOM 2022*. IEEE, 2022, pp. 1509–1518.

[25] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2980–2998, 2010.

[26] E. Wang, M. Zhang, Y. Xu, H. Xiong, and Y. Yang, "Spatiotemporal fracture data inference in sparse urban crowdsensing," in *INFOCOM 2022*. IEEE, 2022, pp. 1499–1508.

[27] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[28] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *NeurIPS 2016*, vol. 29. Curran Associates, Inc., 2016, pp. 1–9.

[29] L. Wan, F. Xia, X. Kong, C.-H. Hsu, R. Huang, and J. Ma, "Deep matrix factorization for trust-aware recommendation in social networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 511–528, 2021.

[30] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *IJCAI 2018*. Morgan Kaufmann, 2018.

[31] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, and P. Li, "DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting," in *PMLR 2022*, vol. 162. PMLR, 2022, pp. 11 906–11 917.

[32] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *WWW 2021*. ACM, 2021, pp. 2069–2080.

[33] J. Ji, J. Wang, C. Huang, J. Wu, B. Xu, Z. Wu, J. Zhang, and Y. Zheng, "Spatio-temporal self-supervised learning for traffic flow prediction," in *AAAI 2023*, vol. 37, no. 4, 2023, pp. 4356–4364.

[34] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *AAAI 2017*, vol. 31, no. 1. AAAI, 2017.

[35] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *AAAI 2019*, vol. 33, no. 1. AAAI, 2019, pp. 5668–5675.

[36] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *SIGKDD 2013*. ACM, 2013, pp. 1436–1444.

[37] L. BAI, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *NeurIPS 2020*, vol. 33. Curran Associates, Inc., 2020, pp. 17 804–17 815.

[38] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *AAAI 2023*, vol. 37, no. 4. AAAI, 2023, pp. 4365–4373.

[39] K. Xie, L. Wang, X. Wang, G. Xie, and J. Wen, "Low cost and high accuracy data gathering in wsns with matrix completion," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1595–1608, 2017.

[40] I. Markovsky, *Low rank approximation: algorithms, implementation, applications*. Springer, 2012, vol. 906.