

Privacy-Preserving Data Completion with Location and Value Obfuscation in Spatial Crowdsourcing

Wenbin Liu, *Member, IEEE*, Hao Du, Lanxin Li, Qi Deng, Liang Wang, Funing Yang, En Wang, *Member, IEEE*, and Jie Wu, *Fellow, IEEE*

Abstract—Spatial-temporal data is essential for various applications. Traditional data collection methods often have limited coverage, while Spatial Crowdsourcing, which leverages mobile devices, offers broader data acquisition but also presents challenges with data sparsity. To infer missing data based on observed data, existing data completion techniques often require detailed spatial-temporal information, raising privacy concerns. Some methods attempt to protect privacy by obfuscating locations; however, this can disrupt the original spatial-temporal correlations, leading to increased errors in data completion. To address this, we propose a differential privacy-based obfuscation strategy that safeguards privacy while maintaining data accuracy. Our approach employs a prediction model to extract an importance matrix and a relationship matrix from historical data. These matrices are utilized to map actual user locations to more beneficial locations for data completion, followed by value adjustments to maintain availability and preserve spatial-temporal correlations. We demonstrate that our mechanism satisfies differential privacy that provides effective privacy guarantees. Evaluations of five real-world datasets show that our obfuscation mechanism outperforms other baseline methods at the same level of privacy protection.

Index Terms—Spatial Crowdsourcing, Location Privacy, Differential Privacy

I. INTRODUCTION

THE advancements in mobile devices and sensing technologies have made Spatial Crowdsourcing [1] an important method for data collection recently, such as environmental monitoring, urban planning, and transportation management [2]–[5]. This approach leverages numerous smart devices, including smartphones and sensors, dispersed throughout society to gather extensive data from the specified time and location, thereby providing better services. However, it is improbable that crowdsourcing users will encompass all spatial-temporal regions completely, leading to data sparsity that necessitates data completion [6]. Significant progress has been made in data completion techniques like matrix factorization [7], compressed sensing [8], and deep learning [9]–[11], which infer missing values based on observed data. These methods rely on accurate spatial-temporal information from the collected data to make inferences, raising serious

Wenbin Liu, Hao Du, Lanxin Li, Yutong Liu, Qi Deng, Funing Yang, and En Wang are with the College of Computer Science and Technology, Jilin University, Changchun 130012, China (e-mail: liuwenbin@jlu.edu.cn; duhao22@mails.jlu.edu.cn; lxli22@mails.jlu.edu.cn; dengqi5522@mails.jlu.edu.cn; yfn@jlu.edu.cn; wangen@jlu.edu.cn).

Liang Wang is with the College of Computer Science, Northwestern Polytechnical University, Xi'an, 710129, China (e-mail: liangwang@nwpu.edu.cn).

Jie Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: jiewu@temple.edu).

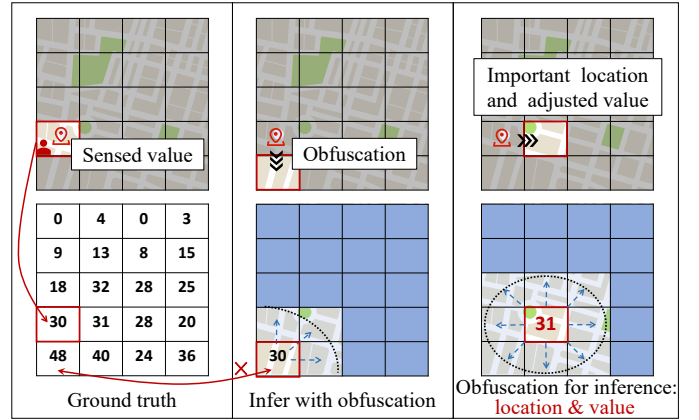


Fig. 1. Completion results of obfuscation mechanism considering different factors. The left figure shows the users' true locations and the corresponding sensed values at different locations. The middle figure illustrates the case where the target region lies in the boundary area. The right figure shows the case where the target location is mapped to a more important location with data adjustment.

privacy concerns. While uploading detailed location and time data can expose sensitive user information, including movement patterns, health status, and lifestyle habits [12], [13]. Therefore, there is a pressing need for privacy-preserving methods that protect user privacy while ensuring accurate data completion [14] in Spatial Crowdsourcing.

A straightforward approach to protect user privacy is to obfuscate the sensed data. Existing privacy-preserving approaches can be roughly divided into several categories [15]. Obfuscation-based methods protect privacy by perturbing or replacing the original locations or sensed values before data release, making them practical for local deployment and real-time reporting. Secure multi-party computation (MPC) enables multiple parties to jointly compute target results without directly revealing their private inputs, but it usually introduces considerable communication and computation overhead [16]. Federated learning (FL) reduces the need to centralize raw data by keeping local data on user devices and only exchanging model updates, but it is mainly designed for collaborative model training rather than directly protecting the released sensing records [17]. Compared with MPC and FL, obfuscation-based methods are more directly applicable to spatial crowdsourcing scenarios, where users need to continuously upload sensing data while protecting sensitive location information. [15] [18] In order to make it difficult for an attacker to determine the actual location of users, these obfuscation techniques typically require randomization, which

will inevitably affect the sensing results. Therefore, traditional methods find it difficult to simultaneously achieve privacy protection and data completion. To conceal the true location or sensed value, these methods typically rely on adding noise or performing randomized obfuscation; however, such operations disrupt the essential spatial-temporal correlations that accurate data completion depends on. As the privacy budget becomes more restrictive, the degree of perturbation increases accordingly, further reducing data utility and leading to significant degradation in completion accuracy. Consequently, under traditional frameworks, privacy and utility exhibit an inherent trade-off, making it nearly impossible to ensure strong privacy protection while maintaining high-quality completion. It is important to note that, under traditional privacy protection frameworks, existing methods mainly focus on obfuscating the original locations, while overlooking the fact that different target locations may provide different levels of information for subsequent data completion. The works [19], [20] state that data from specific locations are more helpful to the accuracy of the completion results. For example, data from the contamination center is obviously more valuable than data from peripheral locations. In Fig. 1, we depict this difference on the right and center sides. Drawing from these realizations, we suggest a logical strategy: obfuscate locations to more advantageous ones and utilize their value to facilitate completion. Some target locations are more suitable for supporting data completion tasks, as they are more capable of preserving useful characteristics, such as the global spatial structure, that are beneficial for data completion. Therefore, we prefer to obfuscate the data to these important locations. Although the exact original location cannot be obtained, the completion model can still benefit from the key structural information preserved at these important locations, thereby achieving better completion performance. For example, in the subfigure, the entire map is partitioned into grid cells. When the true location is obfuscated to a cell on the boundary of the box, its ability to support inference for the surrounding area is limited. In contrast, when the location is obfuscated to any cell near the center of the box, these more representative cells enable the completion model to better recover the values of relevant grids within the entire block, thereby providing greater utility for data completion. The goal of this strategy is to reduce obfuscation error because certain location information can improve data completion.

Due to the necessity of privacy preservation, our first challenge is to consider how to *prioritize the more important information during the obfuscation process*. Generally speaking, a high importance value for a location means its data are strongly correlated in space and time with many other locations, so observing this location can help the completion model more accurately infer missing values in a broad surrounding area. In order to prioritize these beneficial locations while meeting privacy constraints, we must be able to determine which locations are more contributing to the completion and create an appropriate obfuscation strategy. This optimization allows us to improve data completion by using the value on the obfuscated locations.

Although obfuscating locations to more significant locations

can yield valuable information, it also indicates that the uploaded value is inconsistent with the true value of the uploaded location. [21]. Treating the uploaded value as if it were the true value of this (obfuscated) location will inevitably disrupt spatial correlations and reduce data usability. In dynamic environments, this inconsistency can further exacerbate the problem. Some studies have tried to adjust the uploaded values while obfuscating locations to make the obfuscated values approximate the true values at the obfuscated locations [22], [23]. Nevertheless, their modifications are not dynamically changeable over time. If the dynamic correlation of value over time is not taken into account, the temporal consistency and basic availability of data may be jeopardized. Therefore, *creating a privacy-preserving scheme that keeps the obfuscated data availability* becomes our second challenge.

In addition, privacy protection strategies that simultaneously obfuscate location and data can result in high computational complexity. We need to consider not only which important locations can aid in data completion but also how to correct the data after obfuscation. Furthermore, current data completion algorithms are often based on complex models like neural networks, making it challenging yet essential to assess the importance of locations and the exact value. Therefore, *reducing the complexity* of the obfuscation strategy while maintaining its effectiveness is our third challenge.

To address the first challenge, we propose a privacy-preserving data completion framework for Spatial Crowdsourcing that simultaneously obfuscates both location and value. We first utilize differential privacy to protect user privacy by mapping actual locations to other locations, and extract an importance matrix from the middle-layer features of the prediction model. Using this importance matrix, we design a linear programming-based location obfuscation strategy that redirects original locations to more significant ones, hopefully providing more useful information. To address the second challenge, we use historical data to train a prediction model that predicts a spatial-temporal data relationship matrix, enabling us to perform location obfuscation along with dynamic value adjustment, aiming to mitigate the loss in data usability and enhance completion accuracy. We also incorporate feedback from the completion results to better capture these relationships and mine more crucial locations. To address the third challenge, we optimize the linear programming procedure and introduce the diameter 2-critical graph to simplify the constraints of differential privacy, which significantly improves computational efficiency and reduces the complexity of the linear programming algorithm.

The main contributions of our work are as follows:

- We propose a privacy-preserving data completion framework for Spatial Crowdsourcing that simultaneously obfuscates both location and value, ensuring data completion accuracy while protecting user privacy.
- We propose a obfuscation strategy based on differential privacy. Utilising historical data to train a prediction model, we obtain a location importance matrix that directs location obfuscation in the hopes of providing more useful location information. In addition, we simplify the

differential privacy constraints to reduce the complexity of the algorithm.

- We design a strategy based on dynamic value adjustment. The spatial-temporal data relationship matrix that the predictive model predicted is used to modify the uploaded values. By restoring the spatial-temporal consistency of the data, the policy reduces the error caused by obfuscation and improves the availability and complementary accuracy of the data.
- We use five real datasets to evaluate our proposed privacy-preserving framework. The results demonstrate the effectiveness of our algorithm.

We first review recent related work in Section II. In Section III, we introduce our proposed framework, we then present the implementation of the obfuscation algorithm with detail in Section IV. Finally, we conduct experiments in Section V and conclude in Section V and VII.

II. RELATED WORK

A. Data Completion

Due to the cost of data collection, the collected data in Spatial Crowdsourcing is often incomplete or sparse. Consequently, many existing studies have focused on using the sparse data to perform data completion and proposed various algorithms such as compressed sensing [8], matrix factorisation [7], and deep learning-based methods [9].

Considering that sensed data collected from different spatial-temporal regions has significantly different importance for data completion, some researchers have also focused on region selection. Xie et al. [24] proposed an active sensing scheme to reduce sensing and communication costs and ensure high quality data completion. Wang et al. [19] and Liu et al. [20] introduced a reinforcement learning technique to select more important sensing sub-regions. These studies provide multi-perspective solutions for data completion in terms of considering data structure and sense region selection.

Moreover, in Spatial Crowdsourcing, researchers are interested not only in collecting current data but also in predicting future data. Wang et al. [25] proposed a deep learning-enabled industrial sensing and prediction scheme to achieve highly accurate predictions of future moments using sparse data. Liu et al. [26] proposed a model called Spatio-temporal Transformer (ST-Transformer), which exploits the properties of spatial-temporal data to perform data inference and prediction in Spatial Crowdsourcing. However, the aforementioned work overlooks privacy risks. Nonetheless, their proposed region selection and data prediction methods provide a valuable foundation and support for our work.

B. Location Privacy Protection

In the fields of Sparse Crowdsourcing, location-based services (LBS), and the internet of things (IoT), privacy preservation has been extensively researched. Traditional privacy-preserving techniques that use generalization or grouping, like k -anonymity [27], l -diversity [28], and t -closeness [29], lower the risk of re-identification. Differential privacy is progressively emerging as a key method for protecting privacy because

it has rigorous mathematical guarantees, can successfully reduce the impact of adversary prior knowledge, and offers more dependable security [30], [31]. Adding noise to the data is a popular strategy for achieving it. For users with different semantic locations, Min et al. [32] suggested a semantic adaptive geo-indistinguishability mechanism that adds random noise to provide individualized privacy protection. Nevertheless, noise leads to data distortion and decreased usability of spatial-temporal information [33], [34].

Beyond perturbation-based privacy protection, recent studies have also explored the use of secure multi-party computation (MPC) and federated learning (FL) for privacy-preserving tasks. Existing MPC-based works mainly focus on protecting the confidentiality of participants' inputs during collaborative computation, but they typically rely on heavy cryptographic protocols and thus often suffer from considerable communication and computation overhead in large-scale, resource-constrained, or real-time settings [16]. In contrast, FL-based studies mainly aim to reduce privacy exposure by keeping raw data local and further enhance protection through mechanisms such as secure aggregation, differential privacy, or other complementary techniques in distributed training [18]. However, these methods are primarily designed for collaborative modeling or distributed analytics, rather than directly protecting released spatial data itself. Therefore, although MPC and FL are important directions in privacy-preserving research, they are not the most direct solutions for our setting, where the goal is to protect released spatial data while preserving utility for downstream tasks.

Another technique to enhance privacy protection is to intentionally obfuscate the location, so the location uploaded by the user is not the real one but a fake one [35], [36]. However, this approach may lead to a large loss of accuracy of data completion, so methods must navigate the trade-off between protecting privacy and maintaining data utility.

Zhou et al. [37] designed a location obfuscation scheme that maintains relevance of the neighbourhood of non-zero elements. Ren et al. [38] proposed an LBS scheme that protects privacy through Geo-indistinguishability while ensuring the accuracy of users' query service. However, obfuscating only the location without changing the obfuscated values may lead to errors in data inference.

Wang et al. [23] used differential privacy to obfuscate location and proposed a data adjustment function to adjust the value. However, this method aims to reduce the error between the original location and the obfuscated location, which may negatively impact data completion. In this paper, we direct the obfuscation towards critical locations to mitigate the loss in data usability and achieve better completion accuracy compared to conventional obfuscation methods.

III. SYSTEM AND PROBLEM FORMULATION

In this section, we introduce the system model of our proposed privacy-preserving data completion framework in Spatial Crowdsourcing and describe the specific processes involved. Fig. 2 shows the overview of our proposed privacy protecting framework. The main symbols we used and their meanings are shown in TABLE I.

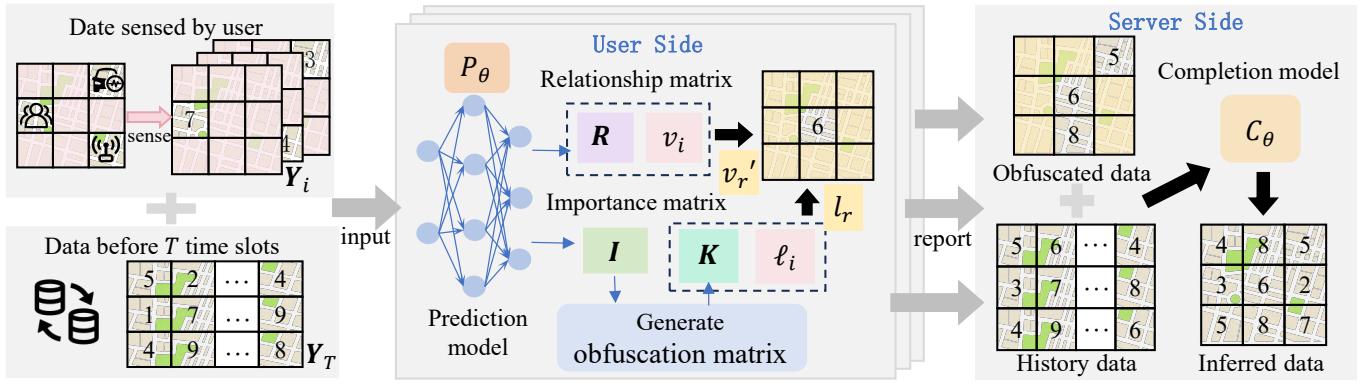


Fig. 2. The Workflow of Our Method. It consists of three parts: data sensing, data obfuscation and data completion.

TABLE I
MAIN NOTATION

Notation	Meaning
M, m	Entire sense map and number of sub-regions
n, s	Number of users and sensed locations
t, T	Current time and T time slices before t
ℓ_i, ℓ_r	True location of u_i and reported location
v_i, v_r	True value of ℓ_i and reported value of ℓ_r
$\mathbf{K}$	Obfuscation probabilistic function
$\mathbf{R}, \mathbf{I}$	Relationship Matrix and Importance Matrix
$\mathbf{Y}_T$	Complete matrix for the previous T time slices
$\mathbf{Y}_i, \mathbf{Y}_i'$	Real sensed / Obfuscated matrix of u_i
$\mathbf{Y}_r, \mathbf{Y}_r'$	Real / Obfuscated matrix received by server
$\mathbf{Y}_c, \mathbf{Y}_c'$	Obtained by completing with $\mathbf{Y}_r / \mathbf{Y}_r'$
$\mathbf{Y}_p$	Complete result predicted by prediction model
P_θ	Prediction model
C_θ	Completion model

A. System Model

Under the Sparse Crowdsourcing context, the privacy-preserving data completion framework has two primary roles, the **user** and the **server** of the sensing system. The primary responsibility of the user is to fill out the sensing task by providing the server with the current location and the associated sensing value. Additionally, because of the high sparsity and limited coverage of the user-uploaded data, inference techniques are typically required in conjunction with the location information carried by the data once the server has collected the data from all users. The primary concern of the sensing system is the accuracy of the completion result. Applications for this scenario are numerous in the domains of environmental monitoring and smart cities, including traffic flow analysis and air quality prediction.

Nonetheless, most users do not have total faith in the server and are concerned that sharing spatial-temporal data may compromise their privacy and, thus, their interests. Reporting real-time traffic flow, for instance, is a perceptual task for the user that can easily result in their own security risks. Therefore, even though there may be additional computational overhead, users would like to be able to lower the risk of privacy leakage through certain privacy-preserving techniques,

like obfuscation or noise addition. This obviously goes against what the server wants in terms of accurate completion results. Privacy protection can effectively lower the risk of sensitive information being leaked, but it also inevitably reduces the quality of uploaded data of user, which in turn affects the server's data-completion performance.

We suggest a privacy-preserving data completion framework in order to resolve this conflict. The goal is to enable users to be locally protected in advance while ensuring that the server can still infer accurate completion results even after accepting the obfuscated data. In practical deployment, the privacy-sensitive obfuscation process is mainly executed on the user side before data uploading, while more computationally intensive tasks such as model training and large-scale completion can be handled on the server or in the cloud.

B. Problem Formulation

We focus on a Spatial Crowdsourcing scenario. We divide the entire sense map M into m sub-regions, and n users sense s sub-regions at each moment. When $s = n$, each user senses one sub-region at each moment. The matrix sensed by user u_i at each time slice is $\mathbf{Y}_i$, where $\mathbf{Y}_i[\ell_i] = v_i$ indicates that the value of location ℓ_i sensed by u_i is v_i , and the rest of the matrix contains 0 indicating that u_i has not sensed value at these locations. However, the data reported to the server by the users is no longer the real location and value, but a fake location ℓ_r and value v_r' processed by the obfuscation policy $f(\cdot)$. Eventually u_i will report the obfuscation matrix $\mathbf{Y}_i'$. $\mathbf{Y}_i'[\ell_r]$ indicates the obfuscated value v_r' of the reported location ℓ_r , and the rest of the matrix is 0. In this case, the not sensed location is indicated by 0. The zero in this case will not be accepted as valid data in later modules.

$$(\ell_r, v_r') = f(\ell_i, v_i). \quad (1)$$

In the obfuscation policy $f(\cdot)$, the prediction model P_θ will generate two matrices: an importance matrix $\mathbf{I}$ and a relationship matrix $\mathbf{R}$, which represent the importance of each location for the data completion and the relationship between the values of each location, respectively. Both matrices constitute predictions for the current time slice. The user gets the reported obfuscated location ℓ_r with the help of $\mathbf{I}$ and adjusts the value based on the $\mathbf{R}$ to get the reported data v_r' .

After collecting all the matrices reported by users at moment t , the server aggregates them into a sensed data matrix $\mathbf{Y}'_r$. The server completes $\mathbf{Y}'_r$ using a trained completion model C_θ , obtaining the complete data $\mathbf{Y}'_c$,

$$\begin{aligned} \mathbf{Y}'_r &= \{\mathbf{Y}'_0, \mathbf{Y}'_1, \dots, \mathbf{Y}'_n\}, \\ \mathbf{Y}'_c &= C_\theta(\mathbf{Y}'_r). \end{aligned} \quad (2)$$

In contrast, the real data $\mathbf{Y}_r$ is aggregated from the real locations and value sensed by users. The data obtained by completing the true location and value is $\mathbf{Y}_c$.

$$\begin{aligned} \mathbf{Y}_r &= \{\mathbf{Y}_0, \mathbf{Y}_1, \dots, \mathbf{Y}_n\}, \\ \mathbf{Y}_c &= C_\theta(\mathbf{Y}_r). \end{aligned} \quad (3)$$

Problem[Sparse Crowdsourcing with Location and Value Privacy Protection]: Our objective is to protect user privacy while maintaining data availability and the accuracy of completion result. Therefore, we aim to minimize the difference between the obfuscated completion results and the completion results using the real data.

$$\min \sum_{i=1}^m |\mathbf{Y}'_c[\ell_i] - \mathbf{Y}_c[\ell_i]|. \quad (4)$$

IV. METHODS

A. Overall Structure

Our proposed model consists of three components: prediction model, obfuscation mechanism, and completion model. Specifically, the user u_i takes data from the environment as the input of the prediction model. The prediction model is stored locally to compute the value relationship matrix $\mathbf{R}$ and the location importance matrix $\mathbf{I}$ at the current moment. Importantly, training of the prediction model can be carried out on the server side. On user devices, only forward inference is required, whose computational and energy overhead is relatively small and therefore acceptable in practice. $\mathbf{I}$ will help the obfuscation mechanism to generate the obfuscation probability matrix $\mathbf{K}$, and the user will select a location to obfuscate the real location according to $\mathbf{K}$. After that, the value reported by the user is adjusted according to the obfuscated location and $\mathbf{R}$. Then, after receiving the reported location and value from all the users, the server integrates them and obtains the completed result through completion model. To ensure that the completion model understands the important locations captured by the prediction model, the prediction model and completion model use the same attention mechanism.

B. Prediction Model

In the field of machine learning, samples with high representativeness often reflect the essential features of the dataset, thus acting as proxies for a set of analogous samples. For example, if two locations have high similarity, then we can infer the other location easily from the available information of one location. Assuming that ℓ_a is a representative location, meaning it has a greater overall similarity to all locations of M , it can offer more representative information. Due to its strong resemblance to many locations of M , choosing ℓ_a as a training sample implies that other similar locations indirectly

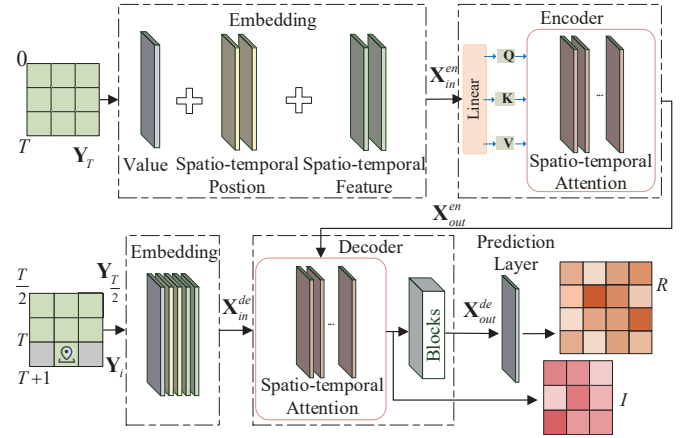


Fig. 3. Workflow of the prediction model

contribute to the model's training [39]. In other words, ℓ_a can provide wealthier information, enhancing the precision and efficacy of the data completion. We refer to this beneficial attribute for completion as location importance.

To obtain this importance, the spatial-temporal correlation of the data needs to be utilized to measure the similarity between different locations. The ST-Transformer proposed by Liu et al. [26] confirms the superiority of the Transformer in capturing spatial-temporal relationships. Therefore we use the Transformer architecture to build prediction model to predict the data relationship and location importance at each moment. The Transformer architecture is kept unchanged to ensure compatibility with the subsequent completion model, allowing the learned relationship and importance matrices to be directly utilized for data completion. The overall workflow of our prediction model is shown in Fig. 3.

Inspired by ST-Transformer, in the embedding layer $\mathcal{Z}(\cdot)$, we add spatial-temporal feature embedding, in addition to traditional data embedding and spatial-temporal position embedding, enabling deeper spatial-temporal sensing and understanding capabilities.

$$\mathcal{Z}(\cdot) = \mathbf{Y}_{val} + \mathbf{T}_{pos} + \mathbf{S}_{pos} + \mathbf{T}_{feat} + \mathbf{S}_{feat}, \quad (5)$$

where $\mathbf{Y}_{val}$ represents the value of the input data, $\mathbf{T}_{pos}$ and $\mathbf{S}_{pos}$ denote the spatial-temporal position embedding, i.e., the current timestamp. Furthermore, $\mathbf{T}_{feat}$ and $\mathbf{S}_{feat}$ stand for temporal and spatial feature embeddings, respectively, which are feature embeddings based on extra feature information such as vacation, coordinates, and so on.

The historical data $\mathbf{Y}_T$ of the previous T time slices is passed through the embedding layer to obtain the input $\mathbf{Y}_{in}^{en}$ of the encoder. In ST-Transformer, the decoder input doesn't go through value embedding. Instead, we use $\mathbf{Y}_{T/2}$ as $\mathbf{Y}^{token}$, $\mathbf{Y}_i$ as $\mathbf{Y}^{target}$. The input to the decoder consists of these two parts. The input of partial historical data $\mathbf{Y}_{T/2}$ can help the decoder capture correlations.

$$\begin{aligned} \mathbf{Y}_{in}^{en} &= \mathcal{Z}(\mathbf{Y}_T), \\ \mathbf{Y}_{in}^{de} &= \mathcal{Z}(\text{Concat}(\mathbf{Y}^{token}, \mathbf{Y}^{target})), \end{aligned} \quad (6)$$

where $\mathbf{Y}_{in}^{en}$ and $\mathbf{Y}_{in}^{de}$ enter the encoder and decoder respectively, and their core structure is the attention mechanism.

Algorithm 1 Prediction Model

Input: Prediction Model P_θ , $\mathbf{Y}_i$, $\mathbf{Y}_T$

Output: $\mathbf{I}$, $\mathbf{R}$

- 1: send $\mathbf{Y}_i$ and $\mathbf{Y}_T$ to **Embedding**
- 2: get $\mathbf{Y}_{in}^{en}$ and $\mathbf{Y}_{in}^{de}$ by Eq. (6)
- 3: send $\mathbf{Y}_{in}^{en}$ to **Encoder**
- 4: calculate attention score by Eq. (7) and (8)
- 5: calculate $\mathbf{Y}_{out}^{en}$
- 6: send $\mathbf{Y}_{in}^{de}$ and $\mathbf{Y}_{out}^{en}$ to **Decoder**
- 7: calculate attention score by Eq. (7) and (8)
- 8: get $\mathbf{Y}_{out}^{de}$
- 9: extract $\mathbf{I}$ from attention output of $\mathbf{Y}_{out}^{de}$
- 10: get $\mathbf{R}$ by inputting $\mathbf{Y}_{out}^{de}$ to prediction layer
- 11: **return** $\mathbf{I}$, $\mathbf{R}$

To help the model extract more information, we use a spatial-temporal attention mechanism in conjunction with the spatial-temporal embedding. Unlike self-attention, we avoid directly calculating the complete attention of all sub-regions, thereby reducing significant complexity. Instead, we first calculate the temporal attention of the sub-regions before computing the spatial attention. This mechanism captures accurate relationships to ensure the prediction performance [26].

For each attention head, there are three learnable weight matrices $\mathbf{W}_i^q$, $\mathbf{W}_i^k$, $\mathbf{W}_i^v$ that project the original data to form three matrices, the query matrix $\mathbf{Q}_i$, the key matrix $\mathbf{K}_i$, and the value matrix $\mathbf{V}_i$. Then the attention score is calculated through $\mathcal{A}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)$.

It defines SMA as the calculation of spatial attention and TMA as the calculation of temporal attention.

$$\begin{aligned} [Q_i^{(T)}, K_i^{(T)}, V_i^{(T)}] &= \mathbf{Y}_{in}[\mathbf{W}_i^{q,(T)}, \mathbf{W}_i^{k,(T)}, \mathbf{W}_i^{v,(T)}]. \\ TMA(Q^{(T)}, K^{(T)}, V^{(T)}) &= Concat(h_1^{(T)}, \dots, h_k^{(T)})\mathbf{W}_T, \\ h_i^{(T)} &= \mathcal{A}(\mathbf{Q}_i^{(T)}, \mathbf{K}_i^{(T)}, \mathbf{V}_i^{(T)}). \end{aligned} \quad (7)$$

We denote the $\mathbf{Y}_{in}^{(T)}$ as the data after temporal attention computation, and calculate the spatial attention:

$$\begin{aligned} [Q_i^S, K_i^S, V_i^S] &= \mathbf{Y}_{in}^{(T)}[\mathbf{W}_i^{q,S}, \mathbf{W}_i^{k,S}, \mathbf{W}_i^{v,S}]. \\ SMA(Q^S, K^S, V^S) &= Concat(h_1^S, \dots, h_k^S)\mathbf{W}_S, \\ h_i^S &= \mathcal{A}(\mathbf{Q}_i^S, \mathbf{K}_i^S, \mathbf{V}_i^S). \end{aligned} \quad (8)$$

The outputs of SMA are extracted to construct the importance matrix $\mathbf{I} \in \mathbb{R}^{m \times 1}$, where $\mathbf{I}[\ell_i]$ indicates the importance of location ℓ_i . This step is made possible by the fact that, when determining the attention score, SMA mainly takes into account how relevant each location is to other locations. In particular, SMA uses the attention mechanism to compute the two-by-two similarity between locations by combining numerical, temporal, and other information. As previously stated, we can consider a location to be more representative, meaning it has greater influence in the data-completion task, when it has a high correlation with multiple locations in the sense map M . Therefore, revealing the degree of correlation between a single location and other locations is the main purpose of the importance matrix $\mathbf{I}$. We construct

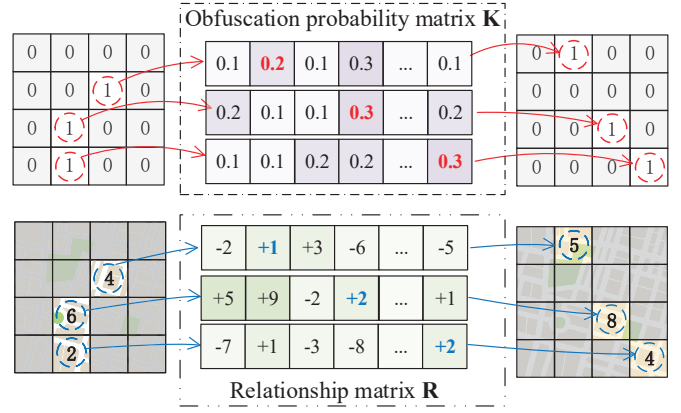


Fig. 4. An example of the obfuscation mechanism

the importance matrix by summing the attention scores of the SMA in order to filter out the more representative locations.

At the same time, using the same attention mechanism, the completion model can understand the spatial correlations captured by the prediction model. Therefore, $\mathbf{I}$ can help the obfuscation mechanism to obfuscate the true location to the one that is considered more important by the completion model. This will reduce the error caused by location obfuscation.

Then $\mathbf{Y}_{out}^{de}$ passes through the other structures like the decoder and the prediction layer, obtains the complete matrix predicted for this moment $\mathbf{Y}_p$, and calculates the value relationship matrix $\mathbf{R}$.

$$\mathbf{R}[\ell_i, \ell_r] = \mathbf{Y}_p[\ell_i] - \mathbf{Y}_p[\ell_r], \quad \forall \ell_i, \ell_r \in \mathcal{L}, \quad (9)$$

where $\mathcal{L} = \{\ell_0, \ell_1, \dots, \ell_{m-1}\}$. Data typically exhibit certain temporal evolution patterns, and these patterns themselves change dynamically over time. The matrix $\mathbf{R}$ characterizes the dependency that should be maintained between two locations at the current time, as determined by these dynamic spatial-temporal patterns, and is used to adjust the values associated with the obfuscated location. The loss function of the prediction model is:

$$loss = \lambda \cdot MSE(\mathbf{R}, \mathbf{R}_{true}) + (1 - \lambda) \cdot MSE(\mathbf{R}, \mathbf{R}'_c), \quad (10)$$

where $\mathbf{R}_{true}$ is the ground truth calculated from the historical data or true data of this moment, and $\mathbf{R}'_c$ is calculated by the output $\mathbf{Y}'_c$ of the completion model. And $\mathbf{Y}'_c$ is the result of completing the obfuscated data $\mathbf{Y}'_r$ reported by all users at the current moment.

To ensure the accuracy of the relationship matrix, we first pre-train the prediction model using historical data, with $\lambda = 1$. We then fine-tune the model using $\mathbf{R}'_c$, helping the prediction model find the locations that are important for the completion task. In order to prevent the accumulation of errors, $\mathbf{R}_{true}$ remains part of the loss function. The algorithm for the prediction model is described in Alg. 1.

C. Obfuscation Mechanism

1) *Normal Mechanism:* At the current moment, the user u_i has collected data from the environment but is reluctant to report the real location ℓ_i and value v_i because it will reveal

sensitive information. For this reason, we need an obfuscation mechanism $P : \ell_i \rightarrow \ell_r$, which allows the user to report the obfuscated location ℓ_r and the corresponding value v_r . This obfuscated location ℓ_r should be such that the attacker cannot recognize the real location of the user, even if he has some prior background knowledge.

To protect privacy better, we apply ϵ -differential privacy [40] as the privacy criterion for the obfuscation mechanism. Rigorous mathematical theories have demonstrated that differential privacy strategies are capable of withstanding attacks based on background knowledge [41].

Definition 1 (ϵ -differential privacy): Given the obfuscation mechanism P that covers the set of locations $\mathcal{L}$, P is ϵ -differential privacy if:

$$\frac{\Pr(P(\ell_i) = \ell_r)}{\Pr(P(\ell'_i) = \ell_r)} \leq e^\epsilon, \quad \forall \ell_i, \ell'_i, \ell_r \in \mathcal{L}, \quad (11)$$

where ϵ is the privacy budget.

We materialize the obfuscation mechanism as an obfuscation probability matrix $\mathbf{K}$, such that $\mathbf{K}(\ell_r | \ell_i)$ indicates the probability of obfuscating location ℓ_i to location ℓ_r . $\mathbf{K}$ satisfies ϵ -differential-privacy if:

$$\frac{\mathbf{K}(\ell_r | \ell_i)}{\mathbf{K}(\ell_r | \ell'_i)} \leq e^\epsilon, \quad \forall \ell_i, \ell'_i, \ell_r \in \mathcal{L}. \quad (12)$$

Meanwhile, for each location ℓ_i , the sum of the probabilities that it obfuscates to other locations should be 1, and the probability cannot be negative. Thus $\mathbf{K}$ should also be satisfied:

$$\sum_{\ell_r \in \mathcal{L}} \mathbf{K}(\ell_r | \ell_i) = 1, \quad \forall \ell_i \in \mathcal{L}, \quad (13)$$

$$\mathbf{K}(\ell_r | \ell_i) \geq 0, \quad \forall \ell_i, \ell_r \in \mathcal{L}. \quad (14)$$

In order to compensate for errors caused by obfuscation, we want the obfuscated location ℓ_r to contribute to data completion. Therefore, we want to obfuscate the true location to a location that is important for completion. Consequently, we use the importance matrix $\mathbf{I}$ extracted from the prediction model, and combine it with the obfuscation matrix $\mathbf{K}$. Therefore, we used the importance matrix $\mathbf{I}$ as weights, with more important locations weighing more, to calculate weighted probability expectation, as follows:

$$\Delta(\mathbf{K}) = \sum_{\ell_i \in \mathcal{L}} \sum_{\ell_r \in \mathcal{L}} \mathbf{I}[\ell_r] \cdot \mathbf{K}(\ell_r | \ell_i). \quad (15)$$

We use linear programming to find an obfuscation matrix $\mathbf{K}$ that maximizes the expectation under differential privacy, taking into account privacy constraints:

$$\begin{aligned} \max_{\mathbf{K}} \Delta(\mathbf{K}) &= \sum_{\ell_i \in \mathcal{L}} \sum_{\ell_r \in \mathcal{L}} \mathbf{I}[\ell_r] \cdot \mathbf{K}(\ell_r | \ell_i), \\ \text{s.t. } &\frac{\mathbf{K}(\ell_r | \ell_i)}{\mathbf{K}(\ell_r | \ell'_i)} \leq e^\epsilon, \quad \forall \ell_i, \ell'_i, \ell_r \in \mathcal{L}, \\ &\sum_{\ell_r \in \mathcal{L}} \mathbf{K}(\ell_r | \ell_i) = 1, \quad \forall \ell_i \in \mathcal{L}, \\ &\mathbf{K}(\ell_r | \ell_i) \geq 0, \quad \forall \ell_i, \ell_r \in \mathcal{L}. \end{aligned} \quad (16)$$

An example of our obfuscation mechanism is shown in Fig. 4. In the upper part of Fig. 4, the matrix on the left indicates the locations sensed by the users, with 1 indicating that the location ℓ_i was sensed. The user probabilistically selects the obfuscated location ℓ_r through the matrix $\mathbf{K}$, which is the location

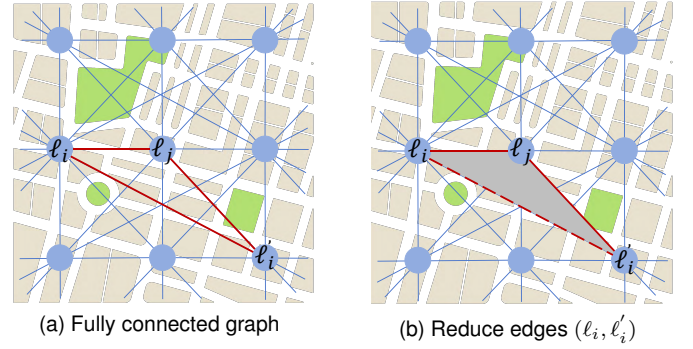


Fig. 5. (a) A division of the map into a square grid. The set of locations contains the centers of the regions. (b) Simplified fully connected graph.

identified by the red probability. After obfuscating the location, the value is also obfuscated. Without obfuscating their value, it means that the value on the obfuscated location is inaccurate, which will also lead to errors in the completion results.

Therefore, we use the relationship matrix $\mathbf{R}$ to adjust the value. $\mathbf{R}[\ell_i, \ell_r]$ denotes the relationship between the values of ℓ_i and ℓ_r , like the blue number in the lower part, which is used to adjust the value v_r reported with ℓ_r . The value collected by user u_i at the real location ℓ_i is v_i , $v'_r = v_i - \mathbf{R}[\ell_i, \ell_r]$.

When the system is deployed in a new environment, the relationship matrix can be set to an all-zero matrix to temporarily disable learning-based guidance, while switching location obfuscation to a standard distance-based mechanism. Under this setting, our method naturally falls back to an obfuscation strategy based on Geographic Indistinguishability. Once sufficient data have been collected, the full method described in the manuscript can be applied.

2) Fast Linear Programming: The computational complexity of existing linear program methods is usually proportional to the number of constraints. Thus, we determine the computational complexity of our obfuscation mechanism by analyzing the complexities of constraints. In particular, when calculating the constraint, it must confirm that the probability of any two locations in the set $\mathcal{L}$ being confused with it satisfies differential privacy for each location $\ell \in \mathcal{L}$. This calculation can be viewed as a graph $G(\mathcal{L}, \mathcal{E})$, each location $\ell \in \mathcal{L}$ represents a vertex. Confirmation of compliance with Eq. (12) is interpreted as traversing the connection between two vertices. That can be represented as the edge set $\mathcal{E}(G)$. Therefore, the fully connected graph must be traversed once for every vertex, with a traversal complexity of $O(|\mathcal{L}|^2)$. As a result, the differential privacy constraint involves pairwise comparisons of locations, leading to a computational complexity of $O(|\mathcal{L}|^3)$. This represents the highest complexity among all the constraints considered, indicating that the mechanism is difficult to calculate for maps with many sub-regions.

To reduce the complexity of the obfuscation mechanism, we introduce diameter 2-critical graph [42], which is used to simplify the number of comparisons between locations in differential privacy. A graph G is said to be diameter 2-critical if the maximum length of the shortest path between any two vertices in the graph G is 2. The following theorem holds:

Theorem 1: If $G(\mathcal{L}, \mathcal{E})$ is a diameter 2-critical graph, and an obfuscation matrix $\mathbf{K}$ satisfies Eq. (17), then $\mathbf{K}$ satisfies

Algorithm 2 Location and Data Obfuscation Framework

Input: ϵ , Prediction Model P_θ , Completion Model C_θ

Output: $\mathbf{Y}'_c$

1: **User Side:**
2: u_i senses $\mathbf{Y}_i$, $\mathbf{Y}_i[\ell_i] \leftarrow v_i$
3: send $\mathbf{Y}_i$ to P_θ
4: P_θ predicts $\mathbf{I}$, $\mathbf{R}$
5: obtain the minimal diameter 2-critical graph $G(\mathcal{L}, \mathcal{E})$
6: calculate the obfuscation matrix $\mathbf{K}$ according to Eq. (19)
7: choose ℓ_r randomly by $\mathbf{K}$
8: $v'_r \leftarrow v_i - \mathbf{R}[\ell_i, \ell_r]$
9: u_i report $\mathbf{Y}'_i$, $\mathbf{Y}'_i[\ell_r] \leftarrow v'_r$
10: **Server Side:**
11: server receives all data reported by users
12: $\mathbf{Y}'_r = \{\mathbf{Y}'_1, \mathbf{Y}'_2, \dots, \mathbf{Y}'_n\}$
13: send $\mathbf{Y}'_r$ to C_θ
14: get completion result $\mathbf{Y}'_c$
15: **return** $\mathbf{Y}'_c$

ϵ -differential-privacy.

$$\frac{\mathbf{K}(\ell_r | \ell_i)}{\mathbf{K}(\ell_r | \ell'_i)} \leq e^{\frac{\epsilon}{2}}, \forall (\ell_i, \ell'_i) \in \mathcal{E}, \forall \ell_r \in \mathcal{L}. \quad (17)$$

Proof: As $G(\mathcal{L}, \mathcal{E})$ is a diameter 2-critical graph, for any two regions $\ell_i, \ell'_i \in \mathcal{L}$, we can find a region ℓ_j and $(\ell_i, \ell_j), (\ell_j, \ell'_i) \in \mathcal{E}$, then for any region $\ell_r \in \mathcal{L}$:

$$\begin{aligned} \mathbf{K}(\ell_r | \ell_i) &\leq e^{\frac{\epsilon}{2}} \cdot \mathbf{K}(\ell_r | \ell_j) \\ \implies \mathbf{K}(\ell_r | \ell_i) &\leq e^{\frac{\epsilon}{2}} \cdot (e^{\frac{\epsilon}{2}} \cdot \mathbf{K}(\ell_r | \ell'_i)) \\ \implies \mathbf{K}(\ell_r | \ell_i) &\leq e^\epsilon \cdot \mathbf{K}(\ell_r | \ell'_i). \end{aligned} \quad (18)$$

The complexity of Eq. (17) is $O(|\mathcal{L}| \cdot |\mathcal{E}|)$. As shown in Fig. 5, a diameter 2-critical graph simplifies the complexity by minimizing the edges that need to be traversed, in contrast to a fully connected graph. We want to find diameter 2-critical graph with the smallest number of edges.

Proposition 1: If $G(\mathcal{L}, \mathcal{E})$ is a diameter 2-critical graph and does not restrict the degree of vertices, $|\mathcal{E}(G)| \geq |\mathcal{L}|$.

Proof: Let $m = |\mathcal{L}|$, $H_d(m, k)$ denotes the set of graphs for which the diameter is $\leq d$ and the degree of vertices is $\leq k$, and $F_d(m, k) = \min_{G \in H_d(m, k)} |\mathcal{E}(G)|$. According to [43], when $d = 2$ and $k = |m - 1|$, $F_2(m, m - 1) = m - 1$, which represents the minimal diameter 2-critical graph, so $|\mathcal{E}(G)| \geq |\mathcal{L}|$. We use this graph thus $O(|\mathcal{E}(G)|) = O(|\mathcal{L}|)$, so the simplified complexity is $O(|\mathcal{L}|^2)$.

Now, the linear programming of the objective and constraints that we use to obtain the obfuscation matrix are:

$$\begin{aligned} \max_{\mathbf{K}} \Delta(\mathbf{K}) &= \sum_{\ell_i \in \mathcal{L}} \sum_{\ell_r \in \mathcal{L}} \mathbf{I}[\ell_r] \cdot \mathbf{K}(\ell_r | \ell_i), \\ \text{s.t. Eq. (13)(14)(17),} \end{aligned} \quad (19)$$

where $G(\mathcal{L}, \mathcal{E})$ satisfies the diameter 2-critical.

D. Completion Model

Our completion model uses the inference architecture of the ST-Transformer, which is similar to the previous Prediction Model in terms of encoder, decoder, and embedding.

However, instead of taking the historical data as part of the input, the completion model directly uses the reported matrix $\mathbf{Y}'_i$ at the current moment by embedding it and using it as the input of the encoder $\mathbf{X}_{in}^{en}$. Therefore the input of the decoder $\mathbf{X}_{in}^{de}$ is the output of the encoder $\mathbf{X}_{out}^{en}$.

The server performs the completion directly after receiving the data $\mathbf{Y}'_r$, so the completion model is a fixed model that has already been trained. We use historical data to train the complementary model in advance. The algorithm for the whole framework is shown in Alg. 2.

Both the completion model and the prediction model are trained using historical data. To capture spatial-temporal features, we use a subset of a dataset as historical data input in the experiments of this paper. These data sets have been anonymized and privacy has been preserved. However, in real-world application scenarios, historical data from the area can be located and utilized following privacy treatment.

V. PERFORMANCE EVALUATION

A. Datasets

To demonstrate the superiority of our method in terms of privacy protection and data utility, we evaluate it on datasets collected from real-world urban sensing and spatial crowdsourcing scenarios. These datasets cover typical spatial-temporal monitoring tasks such as air quality, meteorological indicators, and traffic flow, enabling us to validate the effectiveness of the proposed method in realistic application settings. So we rigorously evaluate our proposed method using five distinct real-world datasets, categorized into three datasets depending on the type.

Air-Quality¹: datasets from air monitoring stations in Beijing, China. It contains hourly recorded data for air components at the different stations. We chose the NO₂ and PM_{2.5} during three weeks as the experimental dataset.

Weather¹: consists of meteorological data, recorded at different air monitoring stations in London, UK, with data recorded hourly on each grid. We chose three weeks of Wind Speed data and Humidity data.

Traffic²: consists of traffic data collected in real time from detectors on California highways. We chose traffic flow as the experimental dataset and selected the top 32 stations.

For all datasets used above, the number of sensors is 32. Unless otherwise specified, all experiments are conducted with 32 sensors.

Although the sensor perceived locations in these datasets are fixed, we sparse the perceived locations during the experiments, meaning that only a few locations are chosen at random for data collection during each training or testing cycle. It can be argued that this method partially accounts for spatial randomness by simulating the dynamic changes in sensed locations.

B. Comparison Algorithms

DU-Min [23]: The goal is to find a location obfuscation strategy that minimizes the uncertainty of value adjustment under differential privacy constraints.

¹<https://www.kdd.org/kdd2018/kdd-cup>

²<http://pems.dot.ca.gov>

TABLE II
COMPLETION PERFORMANCE ON DIFFERENT OBFUSCATION ALGORITHMS

	NO ₂		PM _{2.5}		WindSpeed		Humidity		Traffic	
Method	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
True	0.31990	0.49806	0.17575	0.30670	0.29377	0.44174	0.14144	0.22010	0.22201	0.31341
Average	0.63856	0.88016	0.34395	0.55877	0.53084	0.69876	0.54884	0.72219	0.27383	0.35348
Self	0.61529	0.85274	0.35715	0.58435	0.53659	0.71164	0.48225	0.62414	0.27233	0.35263
Laplace	0.64684	0.88578	0.35466	0.59425	0.57034	0.75391	0.49478	0.63549	0.28151	0.36168
Geo-Ind	0.55326	0.78434	0.32787	0.55654	0.52516	0.71423	0.49905	0.63792	0.27193	0.35325
L-SRR	0.53256	0.73829	0.26879	0.44892	0.53152	0.71223	0.48612	0.62041	0.27089	0.34923
DU-Min	0.57375	0.73616	0.34346	0.50345	0.45816	0.61345	0.44191	0.58923	0.26422	0.34070
Our	<u>0.37090</u>	<u>0.52247</u>	<u>0.19327</u>	<u>0.30676</u>	<u>0.42266</u>	<u>0.55627</u>	<u>0.26373</u>	<u>0.34268</u>	<u>0.23526</u>	<u>0.32033</u>

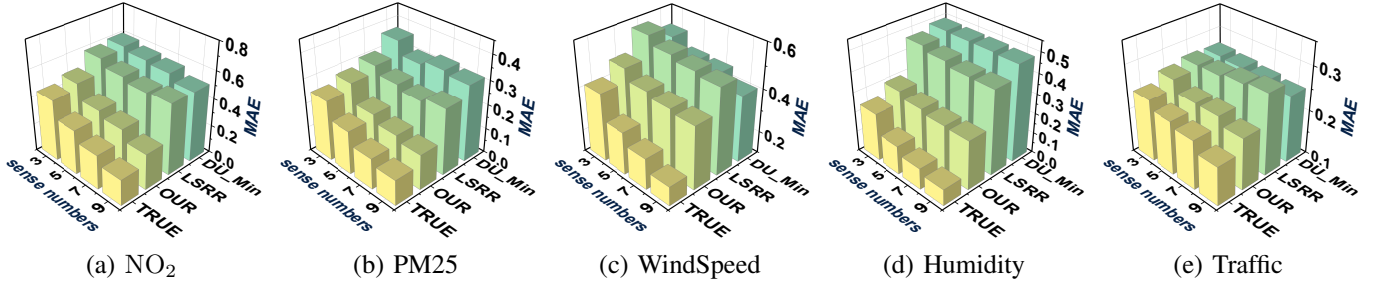


Fig. 6. Completion performance with different sense numbers

TABLE III
COMPLETION PERFORMANCE UNDER MULTIVARIATE OBFUSCATION FOR AIR-QUALITY DATA

Method	True	Average	Self	Laplace
MAE	0.24783	0.49126	0.48622	0.50075
RMSE	0.40238	0.71947	0.71855	0.74002
Method	Geo-Ind	L-SRR	DU-Min	Our
MAE	0.44057	0.40068	0.45861	0.28209
RMSE	0.67044	0.59361	0.61981	0.41462

L-SRR [44]: A new randomization mechanism as the “Staircase Randomized Response” (SRR) in Local Differential Privacy. Specifically, the locations are classified into different sets, and the probability that the true location is obfuscated to each set is based on the distance.

Geo-Ind [41]: Geographic-Indistinguishability is a variant of differential privacy, commonly used to protect location privacy. We apply it to Spat.

Average: The probability of a user choosing any location as the obfuscated location is the same.

Laplace: The probability that users select an obfuscated location as a location follows a Laplace distribution.

Self [45]: This obfuscation algorithm will assign a higher probability to self-obfuscated pairs. That is: $\mathbf{K}(\ell_j | \ell_i) \propto e^\epsilon$, when $\ell_i = \ell_j$, otherwise, $\mathbf{K}(\ell_j | \ell_i)$ is a fixed constant.

We use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to measure the error between the ground truth and the completion result that uses obfuscated locations and value. Since results of the two evaluation metrics are similar, we only show MAE in most experiments for ease of display. All experimental results are derived from repeated experiments. Our experiment platform is equipped with Intel(R) Core(TM) i9-13900K and GeForce RTX 3090.

C. Evaluation Results

In this section, we compare whether our algorithms can guarantee the accuracy of the completion results after obfuscating both locations and value. True means that no obfuscation algorithm is used, and the true locations and values of users are used for completion. We set the number of sensing users to be 5 and the privacy budget $\epsilon = \ln 4$. As shown in TABLE II, for all datasets, our results are always optimal and are closest to True. With the exception of DU-Min, none of the other obfuscation mechanisms adjust the value, leading to large completion errors. For the Average and Laplace methods, they are equivalent to random obfuscation, with the Laplace method satisfying differential privacy. While Self has a higher probability of obfuscating to itself, once it obfuscates to other locations, it results in lower completion quality due to the inaccuracy of the value. Geo_Ind and L-SRR obfuscate the original locations to neighboring locations, resulting in better results due to the general value proximity of the neighboring locations. DU-Min tends to obfuscate to locations where the value error is small, so it sometimes obfuscates to locations that do not improve data completion, resulting in lower performance than our method. We chose DU-Min and L-SRR, which have good results, as the comparison algorithms for the subsequent experiments, and DU-Min is the obfuscation algorithm that takes into account data completion.

We then conducted experiments on the completion performance under different sense numbers n , representing the number of sensing users. As shown in Fig. 6, with increasing number of users, MAE decreases. As more data is sensed, the more accurate the completion results. At the same time, as the number of users increases, the gap between the individual obfuscation algorithms and the results completed using the

true data increases. This is because there is an error between the obfuscated data and the real data. When the number of obfuscation locations increases, the consequent completion error accumulates. On all datasets, our mechanism demonstrates the best results.

Moreover, the dataset used in this experiment is inherently multivariate, and each sensor simultaneously records multiple attributes. TABLE III presents the completion performance under multivariate obfuscation on the Air-Quality dataset. The results show that our method still achieves the best accuracy, obtaining the lowest MAE and RMSE among all mechanisms. This indicates that the proposed approach remains robust even when multiple attributes are perturbed simultaneously. In contrast, DU-Min and Laplace perform worse, because the perturbed values deviate more significantly from the true data distribution, thereby exacerbating completion errors. Geo-Ind and L-SRR exhibit relatively stable performance, which is consistent with the single-variable results in TABLE II, since these mechanisms map values to semantically or spatially neighboring regions. Overall, TABLE III shows that although multivariate obfuscation introduces compounded uncertainty, our method can still maintain high completion accuracy and demonstrates good generalization ability across different perturbation dimensions.

D. Parameter Sensitivity

We experiment with the main parameters of our framework, namely the loss function weight λ of the prediction model and the privacy budget ϵ in the obfuscation mechanism. Through repeating the experiment several times, we determine the average results and delineate the error margins with shading. For clarity, Fig. 8 shows only the margin of error for our method.

1) Weight λ of Loss Function

We experiment with varying weights on different datasets in Fig. 7, the straight line represents the mean value of the MAE and the shaded area represents the confidence interval of the MAE. It can be found that for most datasets, when $\lambda = 0.8$, the MAE of the final result is at the minimum. When $\lambda < 0.8$, the error of the completion results accumulates during the training process of the prediction model, so the importance matrix given by the prediction model is inaccurate. When $\lambda > 0.8$, the prediction model cannot capture the important locations by the completion results, so the obfuscated locations are not optimal for completion, resulting in an increased MAE. Although there are a few datasets where the smallest MAE results from a λ other than 0.8, on the whole, we choose to use $\lambda = 0.8$ for all subsequent experiments.

2) Privacy budget ϵ of obfuscation mechanism

We conducted privacy budget ϵ experiments with values of $\ln 2$, $\ln 4$, $\ln 6$ and $\ln 8$. As shown in Fig. 8, the MAE of our method is the smallest in all datasets with the same privacy budget. As ϵ increases, the errors of all methods decrease. This is because an increase in the privacy budget corresponds to a decrease in privacy protection, which means that the probability will be more biased towards the goal of each obfuscation strategy. In our approach, the probability of obfuscating to a significant location also rises, reducing error. In DU-Min, the probability of obfuscating to the location

with smaller value error rises, leading to better completion. And L-SRR aims to obfuscate locations to closer locations with higher probability. This means that the probability of obfuscating to nearby regions rises when ϵ increases. Since the data has spatial correlation, i.e., the neighboring locations are approximated, the error of the completion decreases. Meanwhile the effect of ϵ on our mechanism is small because the important locations that our obfuscation mechanism tends to obfuscate are constant regardless of how ϵ is changed.

E. Ablation Study

We designed ablation experiments to validate the role of the importance and relationship matrices in our proposed algorithm. For the ablation experiment with the importance matrix, we no longer use the importance matrix to calculate the obfuscation probability matrix in our obfuscation algorithm. Instead, we assume that each location has the same importance. We use this assumption as the objective of the linear programming, $\max_{\mathbf{K}} \Delta(\mathbf{K}) = \sum_{\ell_i \in \mathcal{L}} \sum_{\ell_r \in \mathcal{L}} \frac{1}{m} \cdot \mathbf{K}(\ell_r | \ell_i)$.

For the experiment with the relationship matrix, after locations obfuscation, the values at the obfuscated locations are no longer adjusted. Instead, the original value is used directly, with $v'_r = v_i$.

As shown in Fig. 9, the importance matrix effectively improves the performance of the model. This is because the obfuscation mechanism is able to obfuscate the original locations to important ones, thus improving completion results. The relationship matrix is crucial because it demonstrates that failing to obfuscate values can significantly impair data completion. For some datasets, MAE no longer decreases as the number of users increases after removing the relationship matrix, but instead sometimes rises. This is because when the number of users increases, the amount of inaccurate data also rises. While the sparse ratio decrease is helpful for data completion, it is not enough to offset the negative impact of inaccurate data.

F. Difference on obfuscation probability matrix

In order to show how our obfuscation mechanism differs from others, we plot the obfuscation matrices of DU-Min and our mechanism under the same privacy budget, where red indicates a higher probability of obfuscation. As shown in Fig. 10, DU-Min presents a regular probability matrix, indicating that it will expose information when attacked for a period of long time. On the other hand, our obfuscation matrix obfuscates with an approximate probability except for some important locations with high probability, which makes it difficult for the attacker to find out the real location of the user. This shows that our obfuscation mechanism can achieve the purpose of obfuscating important locations. In addition, compared to DU-Min, our mechanism protects user privacy better because our mechanism will not be attacked by long-term observation attack.

G. Attack Validation

To more thoroughly assess the efficacy of the obfuscation mechanism in protecting privacy, we attempt to mimic the actions of an attacker. On the basis of the uploaded obfuscated location and corresponding value, the attacker attempts to reverse infer the actual location and value of the user, assuming

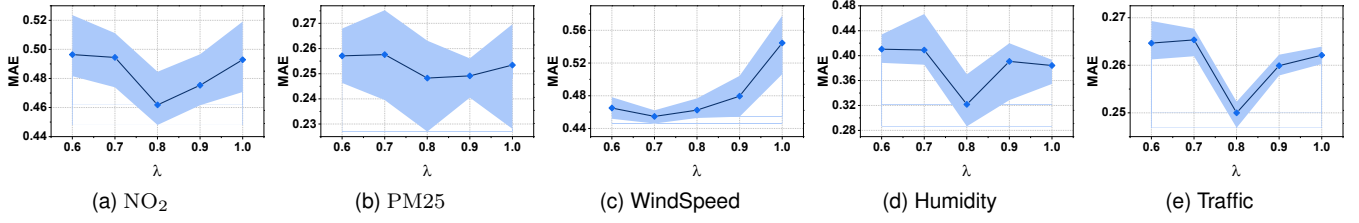


Fig. 7. Loss weight λ on different datasets

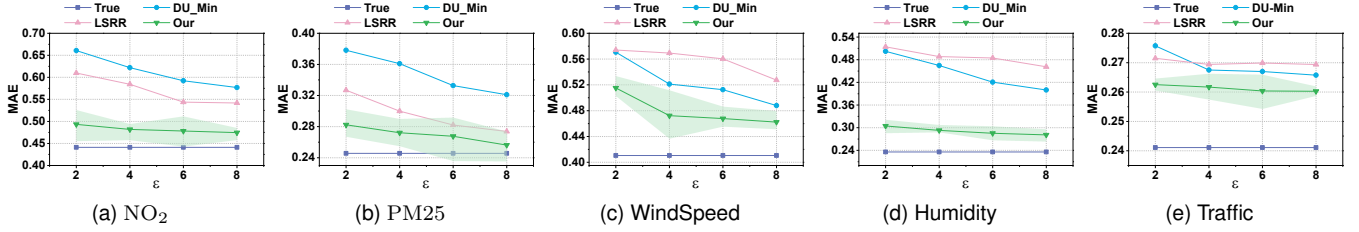


Fig. 8. Privacy budget ϵ on different datasets

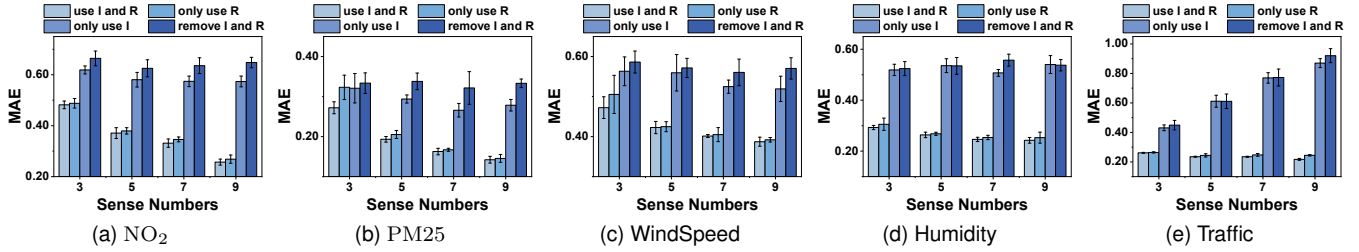


Fig. 9. The effect of the importance matrix and the relationship matrix

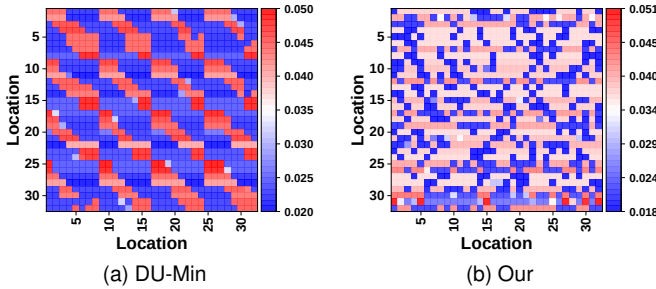


Fig. 10. Obfuscation probability matrix. Increasing the uniformity of the probability distribution makes it more challenging for the attacker to accurately determine the true location.

that he has some background knowledge. In order to compare the privacy-preserving capabilities of various obfuscation algorithms in a more equitable manner, we also test them using the same attack strategy. Since the DU-Min algorithm uses linear programming to adjust data using historical data and generates the obfuscation matrix based on its error. We assume that the attacker can gather sufficient historical data to complete the linear programming process. Using the same obfuscation algorithm, the attacker can then use this information to reconstruct the obfuscation matrix and retrace the uploaded value and obfuscated location. Based on its unique privacy theory, the LSRR algorithm creates the obfuscation matrix. We assume that the attacker is somewhat familiar with this privacy theory

and how it operates. The attacker can also attempt to reverse the actual location and values of the user by mimicking the computation process and obtaining a probability matrix that is equal to the original obfuscation matrix.

In order to test the privacy protection capabilities of our suggested obfuscation mechanism under various attack scenarios, we created two standard attack scenarios: the attacker posing as a user and taking the server hostage. By assuming the identity of a user, the attacker can acquire a model for predicting the importance matrix and is accustomed to the operation of the obfuscation mechanism. The attacker attempts to calculate the obfuscation matrix at that precise moment by obtaining the importance matrix I_{attack} . Meanwhile, the attacker can access the location and value uploaded by other users because we have set here that user-uploaded value and locations are not peer-to-peer encrypted. Thus, by using the obfuscation matrix, the attacker reverses the actual location and value of other users.

The experimental results are shown in Fig. 11. The likelihood that the location attacker restores is the actual location of user is the privacy leakage degree in Fig. 11 (a). The attack results of the Average obfuscation strategy are included for a more intuitive comparison of the privacy protection effect. Its attack attempts to recover the location and value by averaging the probabilities, which is similar to the average obfuscation mechanism. Fig. 11 (b) contains two indicators. The MAE of

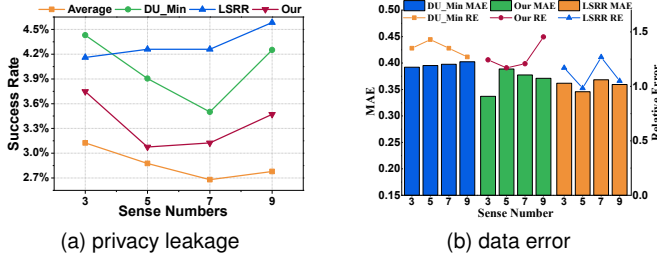


Fig. 11. The effect of an attack disguised as a user (a) Success rate of locations guessed by the attacker (b) Errors in the value guessed by the attacker and their relative proportions to the real value.

TABLE IV
ACCURACY OF THE ATTACKER'S BACKTRACKED LOCATION

Sense Number	Average	DU-Min	LSRR	STDO-Imp	STDO-Rel
3	3.18%	4.45%	4.15%	3.75%	3.85%
5	2.82%	3.95%	4.25%	3.09%	4.23%
7	2.67%	3.55%	4.25%	3.18%	3.38%
9	2.78%	4.25%	4.57%	3.48%	3.92%

the data that the attacker deduced and the actual value is shown in the bar chart. A higher MAE suggests that the obfuscation mechanism is more effective for protecting privacy overall and that the attacker finds it difficult to accurately restore all of the real value. The line graph, on the other hand, shows the Relative Error (RE), or the percentage of the error to the actual value, and can be used to see how closely the restored individual value matches the actual value. A high relative error means that the deviation of attacker in restoring each value is significant. Our algorithm is found to be as effective as other obfuscation algorithms in location attacks, making it difficult for the attacker to determine the user's actual location. Furthermore, while accounting for data availability and preserving the accuracy of data completion, our mechanism still has a lower privacy leakage than other algorithms. Even though the success rate is not as low as the average, We think that the difference between the 4% and 2% values is of the same order of magnitude, which is acceptable in terms of privacy protection. This slight compromise in privacy protection is a reasonable price to pay because our mechanism is now more focused on the accuracy of the completion result, causing the increase in success rate. In addition,, we have the same protective effect as other algorithms when it comes to data errors. This demonstrates how our mechanism can concentrate obfuscation on strategic areas while maintaining user privacy.

The attacker can also compute the relation matrix $\mathbf{R}_{server}$ on its own and obtain a lot of historical data when the server is held hostage. The user has actually evaded the distrusted server by opting to obfuscate locally, and our attention is on the fact that the attacker can obtain a relation matrix based on historical data. It uses user-uploaded data to obtain the completion result, which results in the obfuscated relation matrix $\mathbf{R}_{attack}$. Based on these two matrices, it speculates on optimization and applies the Hungarian algorithm to minimize the error between $\mathbf{R}_{server}$ and $\mathbf{R}_{attack}$ in order to find a globally optimal mapping relation. Fig. 12 displays the results

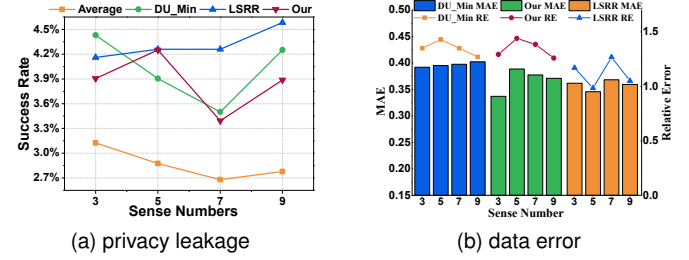


Fig. 12. The effect of hostage server attack (a) Success rate of locations guessed by the attacker (b) Errors in the value guessed by the attacker and their relative proportions to the real value.

TABLE V
COMPUTATION TIME AND MEMORY USAGE

	Time	Memory	Energy
calculate $\mathbf{I}$ and $\mathbf{R}$	0.067 s	28.674 MB	-
Our-Fast	0.618 s	371.125 KB	-
Our-Normal	4.481 s	604.125 KB	-
Local operation	0.660 ms	1.331 MB	8.24e-4 mWh/run

of the experiment.

H. Framework Attack Evaluation

In this section, we attempt to simulate the behavior of attackers. We assume that the attacker possesses certain background knowledge and aims to infer users' true locations and true values from the obfuscated locations and the corresponding reported values uploaded by users.

To ensure fairness, we apply the same attack strategy to all baseline obfuscation algorithms and conduct an in-depth comparison of their privacy protection capabilities. To evaluate privacy protection performance under different conditions, we design two representative attack scenarios: an attacker posing as a normal user and an attacker hijacking the server.

In the scenario where the attacker poses as a user, the attacker can obtain the model used to predict the importance matrix and is familiar with the detailed operation of the obfuscation mechanism, and may attempt to compute the corresponding obfuscation matrix. We further assume that the attacker can observe the locations and data reported by other users. By leveraging this information together with the obfuscation matrix, the attacker attempts to infer the true locations and true values of other users. We denote this attack scenario as STDO-Imp in our experiments.

In the scenario where the attacker hijacks the server, the attacker can access a large amount of historical data and compute the relationship matrix, as well as derive the obfuscated relationship matrix. Based on these two matrices, the attacker performs an optimization procedure and applies the Hungarian algorithm to minimize the error between the known relationship matrix and the obfuscated relationship matrix, thereby identifying a globally optimal mapping between locations. We denote this attack scenario as STDO-Rel in our experiments.

The experimental results are reported in Table IV. In Table IV, the privacy leakage degree is defined as the attacker's success rate in location inference, i.e., the probability that the

TABLE VI
TIME OF CALCULATING OBFUSCATION MATRIX

	NO2	PM25	WindSpeed	Humidity	Traffic
LSRR	0.018 s	0.019 s	0.018 s	0.020 s	0.019 s
DU-Min	3.747 s	4.794 s	4.897 s	4.912 s	4.889 s
Our-Fast	0.633 s	0.639 s	0.618 s	0.626 s	0.631 s

TABLE VII
RUNTIME COMPARISON OF THE OPTIMIZED LP AND THE TRADITIONAL LP UNDER DIFFERENT NUMBERS OF LOCATIONS

Number of locations	Optimized LP	Traditional LP
20	0.062 s	0.323 s
30	0.106 s	2.277 s
40	0.175 s	12.614 s
60	0.819 s	137.056 s
80	0.939 s	—
100	1.913 s	—

inferred location matches the user's true location. To more intuitively demonstrate the privacy protection effect, we also include the Average obfuscation strategy as a baseline, since it is equivalent to fully random obfuscation and does not allow the attacker to leverage any background knowledge to improve inference.

It can be observed that, while fully accounting for data utility and ensuring completion accuracy, our method still achieves a lower privacy leakage degree than other methods. Although its inference success rate is slightly higher than that under the Average strategy, we argue that the difference between 4% and 2% is of the same order of magnitude and remains acceptable from a privacy protection perspective. It is worth noting that this slight increase in inference success rate stems from the fact that our mechanism places greater emphasis on the accuracy of data completion results. We consider this minor degradation in privacy to be an acceptable cost for achieving better data completion performance.

I. Computation time and memory usage

In practical deployment, the computational overhead of our method can be divided into an offline/cloud-side stage and an online/device-side stage. The cloud-side stage mainly includes prediction model training and the computation of the obfuscation matrix $\mathbf{K}$, both of which are completed offline. The experimental platform is equipped with an Intel(R) Core(TM) i9-13900K and a GeForce RTX 3090. In contrast, the mobile device only needs to execute a lightweight online stage, including model inference and sampling based on the precomputed obfuscation matrix. Therefore, the cloud side has substantially stronger computational resources than ordinary mobile devices, and its execution speed can be significantly faster than that of the local test smartphone we currently use.

Table V reports the computation time and memory usage for a single user. The first three rows correspond to cloud-side computation. Compared with the traditional linear programming method based on a fully connected graph, the fast linear programming method based on a diameter-2 critical graph significantly reduces both time and memory overhead, demonstrating that our design effectively lowers the resource

consumption of the obfuscation mechanism. Although the computation of $\mathbf{I}$ and $\mathbf{R}$ introduces additional cost, Fig. 9 shows that they significantly improve the completion performance, and thus this overhead is justified.

To verify the practical feasibility of the device-side stage, we exported the prediction model to ONNX format and conducted local tests on a Pixel 9a (Android API 35). The results show that the on-device inference latency is only 0.660 ms per run, with very small memory fluctuation. In addition, the estimated energy consumption is only 0.082438 mWh for 100 runs, indicating that the battery overhead of the local stage is very limited in practice. Since the computationally intensive model training and obfuscation-matrix construction are performed on the cloud side, typical smartphones can readily support the proposed method.

Additionally, as indicated in TABLE VI, we contrasted the amount of time spent on the obfuscation mechanism with that of other algorithms. There is no denying that protecting privacy requires the use of additional resources, such as more processing time. In this sense, L-SRR has a lower computational overhead since it computes the obfuscation matrix based only on the distances between various positions, which is a fairly straightforward mathematical operation. However, the fact that L-SRR ignores the impact of obfuscation on data completion makes its drawbacks very evident. In contrast, our method and DU-Min will use a little more for computation. This is because, when calculating the confusion matrix, we include an extra factor of location importance. The results of earlier experiments demonstrate that this method successfully lessens the effect of obfuscation on data usability. To attain more accurate data completion, this extra overhead is a necessary trade-off. It is noteworthy that the algorithm's execution time overhead is within an acceptable range, at approximately 0.6 seconds. The entire obfuscation procedure is executed in parallel on the user side, rather than sequentially on the server. Therefore, the additional 0.6-second processing time refers to the per-user local latency for generating an obfuscated report, rather than an accumulated server-side cost. Since each user performs this computation independently and in parallel, the overhead does not scale linearly with the number of participants in large-scale crowdsourcing scenarios. Furthermore, this overhead is reasonable because the real impact of data uploading delays on the system and the user is typically minimal in Sparse MCS.

However, for larger-scale deployment scenarios, it is also necessary to discuss how the runtime of the linear-programming-based obfuscation stage changes as the problem size increases. We conducted additional experiments, and the corresponding results are shown in Table VII. The conventional linear programming method already requires about 10 seconds when the number of locations reaches 40, whereas the optimized linear programming method does not reach this threshold until the number of locations exceeds one hundred. Considering that the datasets used in this paper contain 32 sensors unless otherwise specified, this indicates that the optimized formulation is already sufficient for most task settings studied in this paper, while the conventional formulation becomes inefficient at a smaller scale.

In addition, we conducted extra local tests on city-scale settings involving more than one thousand locations, where the obfuscation time of the optimized method was on the order of minutes. Since the obfuscation process itself can be further deployed in the cloud, its runtime can be expected to decrease further in a higher-performance cloud computing environment. For city-scale scenarios involving hundreds or even thousands of locations, the overall runtime trend further indicates that the gap between the two formulations becomes increasingly pronounced as the number of locations grows, while the optimized formulation still demonstrates better practical scalability.

VI. DISCUSSION

A. Discussion on the use of multivariate data

Although all experiments in this paper are conducted on scalar time series (e.g., a single pollutant or traffic indicator at each location), many recent studies have shown that spatial-temporal models can effectively handle multivariate inputs, such as multiple pollutants and meteorological factors for air quality forecasting, or multiple traffic-related variables for traffic flow prediction. For example, Du et al. design a hybrid CNN-BiLSTM framework that explicitly learns the interdependence of multivariate air quality time series including PM_{2.5}, wind speed, temperature, and other factors [46]. In the traffic domain, multivariate or multi-feature spatial-temporal models have been developed to jointly model correlated variables such as flow, speed, and occupancy, or to integrate multiple traffic features within graph-based architectures [47]. Motivated by these studies, our framework can be extended by treating each report as a vector and adopting a multivariate prediction or completion backbone to learn the importance matrix and a multivariate relationship matrix. We can then perform location obfuscation based on the importance matrix, and conduct value adjustment based on the relationship matrix, so that privacy protection and data utility can still be jointly maintained in multivariate settings.

B. Privacy budget in multi-round reporting

Building on the above discussion of extending our method from univariate to multivariate data, we further consider its deployment in new environments. In practice, our prediction model P_θ is trained from historical data. Therefore, in new regions or new tasks where historical data are sparse or unavailable, the system may not be able to reliably capture the underlying importance and relationship patterns, and thus needs to fall back to simpler obfuscation strategies, such as random obfuscation based on Geographic Indistinguishability. To improve practicality under cold-start scenarios, the system can also employ transfer learning and federated learning to accelerate model adaptation. Another option is to initialize the system with a generally pre-trained model and then gradually fine-tune it using newly collected data as it becomes available.

For applications that require continuously reporting real-time locations to support task allocation, a commonly used practical approach, namely location anonymity, can be adopted to help protect privacy [48]. For example, in crowdsourced food-delivery platforms, riders need to continuously report their locations so that the platform can assign tasks such

as order pickup and delivery. If these location reports are exposed over a long period of time, riders' trajectories and sensitive locations may gradually be inferred. In this case, a commonly used location-anonymity mechanism is as follows: when displaying nearby riders, the platform shows real-time GPS coordinates but does not disclose the riders' real identities. Instead, a temporary or fake ID is used for each location report, which is equivalent to assigning a randomly changing pseudonym to each report, thereby protecting the riders' privacy.

At the same time, location anonymity cannot completely eliminate privacy risks. Attackers may still be able to reconstruct trajectories with relatively high accuracy over short time periods. However, it can significantly hinder long-term tracking. Therefore, how to combine cross-time privacy-budget management with identity-decoupling mechanisms such as location anonymity—while also considering the residual privacy risks of location anonymity itself—remains an important and promising direction for future research.

C. Obfuscation strategies against attacks

Our current framework mainly studies the trade-off between privacy protection and data utility, without explicitly modeling a concrete adversarial inference strategy. A promising extension is to incorporate attack-aware objectives into the obfuscation process. For example, under a trajectory/linkage inference attack, the attacker may exploit temporal continuity or spatial transition consistency to associate multiple released reports from the same user. In this case, the obfuscation mechanism could be designed to reduce cross-time linkability, such as by discouraging highly consistent mappings across consecutive releases. Another possible threat is a semantic/prior-knowledge inference attack, where the attacker combines released data with auxiliary information, such as semantic labels, maps, or historical statistics, to infer sensitive places. In such a setting, the obfuscation process could be extended to suppress distinguishable semantic cues while retaining the key spatial structures needed by downstream completion. Exploring these attack-aware extensions is an important direction for future work.

VII. CONCLUSION

In conclusion, this study proposes a framework for obfuscating locations and value of users simultaneously, ensuring accuracy of data completion while protecting user privacy. The framework consists of a prediction model, obfuscation mechanism, and completion model. The obfuscation mechanism uses the importance of the locations from the prediction model to obfuscate locations to significant locations. At the same time, the predicted spatial-temporal relationships are used to adjust the value to reduce the completion error. The obfuscation mechanism follows differential privacy and provides strong privacy protection. The experiments have been conducted on five real-world datasets in three areas: air quality, traffic flow, and weather. The results verify the effectiveness of our proposed framework. However, all datasets used in this work are scalar because the model was not built to handle non-scalar data, like

images. This indicates that the framework's ability to handle increasingly complicated data types is limited. Future research will focus on extending the model to accommodate non-scalar data, further enhancing its applicability in diverse scenarios.

REFERENCES

- [1] L. Kazemi and C. Shahabi, "Geocrowd: enabling query answering with spatial crowdsourcing," in *Proceedings of the International Conference on Advances in Geographic Information Systems*, New York, NY, USA, 2012, p. 189–198.
- [2] W. Liu, Y. Yang, E. Wang, and J. Wu, "User recruitment for enhancing data inference accuracy in sparse mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1802–1814, 2020.
- [3] M. M. Alam, L. Torgo, and A. Bifet, "A survey on spatio-temporal data analytics systems," *ACM Computing Surveys*, vol. 54, no. 10s, pp. 1–38, 2022.
- [4] H. Gong, B. Huang, B. Jia, and H. Dai, "Modeling power consumptions for multirotor uavs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 6, pp. 7409–7422, 2023.
- [5] H. Gong, B. Huang, and B. Jia, "Energy-efficient 3-d uav ground node accessing using the minimum number of uavs," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 12 046–12 060, 2024.
- [6] Y. Zhao and Q. Han, "Spatial crowdsourcing: current state and future directions," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 102–107, 2016.
- [7] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Mathematical Programming Computation*, vol. 4, pp. 333–361, 2012.
- [8] B.-W. Chen, K.-L. Hou, P.-H. Wu, W.-C. Ye, and J.-Y. Huang, "Heterogeneous multiview crowdsensing based on half quadratic optimization for the visual internet of things," *IEEE Wireless Communications*, vol. 28, no. 4, pp. 19–25, 2021.
- [9] J. Fan and J. Cheng, "Matrix completion by deep matrix factorization," *Neural networks : the official journal of the International Neural Network Society*, vol. 98, pp. 34–41, 2018.
- [10] W. Liu, H. Du, E. Wang, J. Lv, W. Liu, B. Yang, and J. Wu, "Spatio-temporal pyramid-based multi-scale data completion in sparse crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 25, no. 1, pp. 1270–1286, 2026.
- [11] W. Liu, H. Du, E. Wang, D. Luan, B. Yang, Y. Yang, and J. Wu, "Non-aligned multi-scale data completion for sparse mobile crowdsensing," in *IEEE/ACM International Symposium on Quality of Service*, 2025, pp. 1–10.
- [12] D. McGraw and K. D. Mandl, "Privacy protections to encourage use of health-relevant digital data in a learning health system," *NPJ digital medicine*, vol. 4, no. 1, p. 2, 2021.
- [13] L. Yu, S. Zhang, Y. Meng, S. Du, Y. Chen, Y. Ren, and H. Zhu, "Privacy-preserving location-based advertising via longitudinal geo-indistinguishability," *IEEE Transactions on Mobile Computing*, vol. 23, no. 8, pp. 8256–8273, 2024.
- [14] J. An, S. Wu, X. Gui, X. He, and X. Zhang, "A blockchain-based framework for data quality in edge-computing-enabled crowdsensing," *Frontiers of Computer Science*, vol. 17, no. 4, 2022.
- [15] H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar, "Location privacy-preserving mechanisms in location-based services: A comprehensive survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 1, pp. 1–36, 2021.
- [16] I. Gamiz, C. Regueiro, O. Lage, E. Jacob, and J. Astorga, "Challenges and future research directions in secure multi-party computation for resource-constrained devices and large-scale computations," *International Journal of Information Security*, vol. 24, no. 1, p. 27, 2025.
- [17] J. C. Zhao, S. Bagchi, S. Avestimehr, K. S. Chan, S. Chaterji, D. Dimitriadis, J. Li, N. Li, A. Nourian, and H. R. Roth, "Federated learning privacy: Attacks, defenses, applications, and policy landscape-a survey," *arXiv preprint arXiv:2405.03636*, vol. 2, 2024.
- [18] S. Saha, A. Hota, A. K. Chattopadhyay, A. Nag, and S. Nandi, "A multifaceted survey on privacy preservation of federated learning: progress, challenges, and opportunities," *Artificial Intelligence Review*, vol. 57, no. 7, p. 184, 2024.
- [19] L. Wang, W. Liu, D. Zhang, Y. Wang, E. Wang, and Y. Yang, "Cell selection with deep reinforcement learning in sparse mobile crowdsensing," in *IEEE International Conference on Distributed Computing Systems*, 2018, pp. 1543–1546.
- [20] W. Liu, L. Wang, E. Wang, Y. Yang, D. Zeglache, and D. Zhang, "Reinforcement learning-based cell selection in sparse mobile crowdsensing," *Computer Networks*, vol. 161, no. C, p. 102–114, 2019.
- [21] B. Niu, Y. Chen, Z. Wang, F. Li, B. Wang, and H. Li, "Eclipse: Preserving differential location privacy against long-term observation attacks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 1, pp. 125–138, 2022.
- [22] L. Wang, D. Zhang, D. Yang, B. Y. Lim, and X. Ma, "Differential location privacy for sparse mobile crowdsensing," in *IEEE International Conference on Data Mining*, 2016, pp. 1257–1262.
- [23] L. Wang, D. Zhang, D. Yang, B. Y. Lim, X. Han, and X. Ma, "Sparse mobile crowdsensing with differential and distortion location privacy," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2735–2749, 2020.
- [24] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, and D. Zhang, "Active sparse mobile crowd sensing based on matrix completion," in *Proceedings of the International Conference on Management of Data*, 2019, p. 195–210.
- [25] E. Wang, M. Zhang, X. Cheng, Y. Yang, W. Liu, H. Yu, L. Wang, and J. Zhang, "Deep learning-enabled sparse industrial crowdsensing and prediction," *IEEE Transactions on Industrial Informatics*, pp. 6170–6181, 2021.
- [26] E. Wang, W. Liu, W. Liu, C. Xiang, B. Yang, and Y. Yang, "Spatiotemporal transformer for data inference and long prediction in sparse mobile crowdsensing," in *IEEE Conference on Computer Communications*, 2023, pp. 1–10.
- [27] Z. Gao, Y. Huang, L. Zheng, H. Lu, B. Wu, and J. Zhang, "Protecting location privacy of users based on trajectory obfuscation in mobile crowdsensing," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6290–6299, 2022.
- [28] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkatasubramanian, "L-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, p. 3–es, Mar. 2007.
- [29] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *IEEE International Conference on Data Engineering*, 2007, pp. 106–115.
- [30] X. Chen, X. Wu, X. Wang, W. Zhao, and W. Xue, "Real-location reporting based differential privacy trajectory protection for mobile crowdsensing," in *International Conference on Big Data Computing and Communications*, 2019, pp. 142–150.
- [31] M. Wang, H. Jiang, P. Zhao, J. Li, J. Liu, G. Min, and S. Dustdar, "Ropriv: Road network-aware privacy-preserving framework in spatial crowdsourcing," *IEEE Transactions on Mobile Computing*, vol. 23, no. 3, pp. 2351–2366, 2024.
- [32] M. Min, H. Zhu, S. Li, H. Zhang, L. Xiao, M. Pan, and Z. Han, "Semantic adaptive geo-indistinguishability for location privacy protection in mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 6, pp. 9193–9198, 2024.
- [33] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Preserving privacy in gps traces via uncertainty-aware path cloaking," in *Proceedings of the ACM Conference on Computer and Communications Security*, 2007, p. 161–171.
- [34] Z. Wang, K. Liu, J. Hu, J. Ren, H. Guo, and W. Yuan, "Attrleaks on the edge: Exploiting information leakage from privacy-preserving co-inference," *Chinese Journal of Electronics*, vol. 32, no. 1, pp. 1–12, 2023.
- [35] P. Pappachan, V. S. Hunsur Manjunath, C. Qiu, A. Squicciarini, and H. Onweller, "Corgi: An interactive framework for customizable and robust location obfuscation," in *IEEE International Conference on Data Engineering*, 2023, pp. 3667–3670.
- [36] L. Wang, D. Yang, X. Han, D. Zhang, and X. Ma, "Mobile crowdsourcing task allocation with differential-and-distortion geo-obfuscation," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 967–981, 2021.
- [37] T. Zhou, Z. Cai, B. Xiao, L. Wang, M. Xu, and Y. Chen, "Location privacy-preserving data recovery for mobile crowdsensing," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, 2018.
- [38] Y. Ren, X. Li, Y. Miao, R. H. Deng, J. Weng, S. Ma, and J. Ma, "Distpreserv: Maintaining user distribution for privacy-preserving location-based services," *IEEE Transactions on Mobile Computing*, vol. 22, no. 6, pp. 3287–3302, 2023.
- [39] D. Hao, E. Wang, W. Liu, W. Liu, J. Yuan, Y. Yang, and J. Wu, "Active sensing for transformer model in sparse mobile crowdsensing," in *IEEE International Conference on Mobile Ad Hoc and Smart Systems*, 2023, pp. 186–194.
- [40] C. Dwork, "Differential privacy," in *International Colloquium on Automata, Languages and Programming*, 2006.

- [41] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the ACM SIGSAC Conference on Computer & Communications Security*, 2013, p. 901–914.
- [42] L. Caccetta and R. Häggkvist, "On diameter critical graphs," *Discrete Mathematics*, vol. 28, no. 3, p. 223–229, 1979.
- [43] P. Erdős, "On a problem in graph theory," *The Mathematical Gazette*, vol. 47, no. 361, pp. 220–223, 1963.
- [44] H. Wang, H. Hong, L. Xiong, Z. Qin, and Y. Hong, "L-srr: Local differential privacy for location-based services with staircase randomized response," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2022, p. 2809–2823.
- [45] S. Agrawal and J. Haritsa, "A framework for high-accuracy privacy-preserving mining," in *IEEE International Conference on Data Engineering*, 2005, pp. 193–204.
- [46] S. Du, T. Li, Y. Yang, and S.-J. Horng, "Deep air quality forecasting using hybrid deep learning framework," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2412–2424, 2019.
- [47] S. Wang, M. Zhang, H. Miao, Z. Peng, and P. S. Yu, "Multivariate correlation-aware spatio-temporal graph convolutional networks for multi-scale traffic prediction," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 3, pp. 1–22, 2022.
- [48] L. Zhou, R. Yang, L. Ma, G. Chen, H. Zhu, L. Yang, Q. Pei, and Q. Li, "The feasibility of location anonymity: An empirical study towards a real-world location privacy protection system in takeout services," in *IEEE INFOCOM 2025-IEEE Conference on Computer Communications*. IEEE, 2025, pp. 1–10.



Wenbin Liu (Member, IEEE) received the B.S. degree in physics and the Ph.D. degree in computer science and technology from Jilin University, China, in 2012 and 2020, where he is currently an associate professor with the College of Computer Science and Technology. He was also a visiting PhD student with the Wireless Networks and Multimedia Services Department, Telecom SudParis/Institut Mines-Telecom, France. His research interests include mobile crowdsensing, spatial-temporal crowdsourcing, and ubiquitous computing.



Hao Du received his B.E. degree in software engineering from Inner Mongolia University, Hohhot, China, in 2022. He enrolled in a combined Master's and Ph.D. program in computer science and technology at Jilin University, Changchun, China, in 2022, where he is currently pursuing the Ph.D. degree. His current research focuses on mobile crowdsensing, spatial-temporal data processing and multi-scale data processing.



Lanxin Li received her B.E. degree in computer science and technology from Harbin Engineering University, Harbin, China in 2022. She is currently studying for the Master's degree in computer system architecture with the College of Computer Science and Technology, Jilin University, Changchun, China. Her current work focus on Sparse Mobile Crowd-Sensing, privacy protection and data completion.



Qi Deng is currently pursuing a bachelor's degree in Computer Science and Technology at Jilin University in China. His research interests include Big Data, Location Privacy, Data Mining, Crowd Intelligence and Mobile Computing.



Liang Wang (Member, IEEE) received the Ph.D. degree in computer science from Shenyang Institute of Automation (SIA), Chinese Academy of Sciences, Shenyang, China, in 2014. He is currently an Associate Professor with Northwestern Polytechnical University, Xi'an, China. His research interests include ubiquitous computing, mobile crowdsensing, and data mining.



Funing Yang received her B.E. degree in Software Engineering from Jilin University in 2010, her M.S. degree in Computer Science from Beijing University of Posts and Telecommunications in 2013, and her Ph.D. degree in Management from Jilin University in 2022. She is currently an Associate Professor at Jilin University, China. Her research interests include mobile CrowdSensing, mobile computing, and massive data mining and integration. She has published several papers in top journals and conferences.



En Wang (Member, IEEE) received his B.E. degree in software engineering from Jilin University, Changchun in 2011, and his M.E. degree and Ph.D. in computer science and technology from Jilin University, Changchun in 2013 and 2016. He was also a joint Ph.D. student with the department of Computer and Information Science, Temple University. He is currently a professor in the Department of Computer Science and Technology at Jilin University, Changchun. His current research focuses on Mobile Computing, Crowd Intelligence and Data Mining.



Jie Wu (Fellow, IEEE) is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu was an IEEE Computer Society Distinguished Visitor and ACM Distinguished Speaker. He is a China Computer Federation (CCF) Distinguished Speaker and a Fellow of the IEEE. Currently, he is a chief scientist at China Telecom and is the director of Cloud Computing Research Institute.