SPACE-TA: Cost-Effective Task Allocation Exploiting Intradata and Interdata Correlations in Sparse Crowdsensing

LEYE WANG, Hong Kong University of Science and Technology
DAQING ZHANG, Key Lab of High Confidence Software Technologies, Peking University
DINGQI YANG, University of Fribourg
ANIMESH PATHAK, myKaarma Labs
CHAO CHEN, Chongqing University
XIAO HAN, Shanghai University of Finance and Economics
HAOYI XIONG, Missouri University of Science and Technology
YASHA WANG, Peking University

Data quality and budget are two primary concerns in urban-scale mobile crowdsensing. Traditional research on mobile crowdsensing mainly takes sensing coverage ratio as the data quality metric rather than the overall sensed data error in the target-sensing area. In this article, we propose to leverage spatiotemporal correlations among the sensed data in the target-sensing area to significantly reduce the number of sensing task assignments. In particular, we exploit both intradata correlations within the same type of sensed data and interdata correlations among different types of sensed data in the sensing task. We propose a novel crowdsensing task allocation framework called *SPACE-TA (SPArse Cost-Effective Task Allocation)*, combining compressive sensing, statistical analysis, active learning, and transfer learning, to dynamically select a small set of subareas for sensing in each timeslot (cycle), while inferring the data of unsensed subareas under a probabilistic data quality guarantee. Evaluations on real-life temperature, humidity, air quality, and traffic monitoring datasets verify the effectiveness of SPACE-TA. In the temperature-monitoring task leveraging intradata correlations, SPACE-TA requires data from only 15.5% of the subareas while keeping the inference error below 0.25°C in 95% of the cycles, reducing the number of sensed subareas by 18.0% to 26.5% compared to baselines. When multiple tasks run simultaneously, for example, for temperature and humidity monitoring, SPACE-TA can further reduce ~10% of the sensed subareas by exploiting interdata correlations.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing**;

This research is partially supported by NSFC Grant nos. 61572048, 71601106, and 61602067, State Language Commission Key Program Grant no. ZDI135-18, Hong Kong ITF Grant no. ITS/391/15FX, the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement 683253/GraphInt), and Missouri S&T Startup Funding for Smart Cyber-Physical Systems Cluster Hire.

Authors' addresses: L. Wang, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, China; email: wly@cse.ust.hk; D. Zhang and Y. Wang, School of Electronic Engineering and Computer Science, Peking University, 5 Yiheyuan Rd, Beijing, China; emails: {dqzhang, wangys}@sei.pku.edu.cn; D. Yang, eXascale Infolab, University of Fribourg, Bd de Pérolles 90, 1700 Fribourg, Switzerland; email: dingqi@exascale.info; A. Pathak, myKaarma Labs, 2698 Junipero Ave #201b, Signal Hill, CA 90755, USA; email: animesh.pathak@mykaarma.com; C. Chen, College of Computer Science, Chongqing University, 174 Shazhengjie, Chongqing, 400044, China; email: cschaochen@cqu.edu.cn; X. Han (Corresponding author), School of Information Management and Engineering, Shanghai University of Finance and Economics, 777 Guoding Rd, Shanghai, China; email: xiaohan@mail.shufe.edu.cn; H. Xiong, Department of Computer Science, Missouri University of Science and Technology, 1201 N State St, Rolla, MO 65409, USA; email: xiongha@mst.edu.

Additional Key Words and Phrases: Crowdsensing, task allocation, data quality

ACM Reference format:
Leye Wang, Daqing Zhang, Dingqi Yang, Animesh Pathak, Chao Chen, Xiao Han, Haoyi Xiong, and Yasha Wang. 2017. SPACE-TA: Cost-Effective Task Allocation Exploiting Intradata and Interdata Correlations in Sparse Crowdsensing. *ACM Trans. Intell. Syst. Technol.* 9, 2, Article 20 (October 2017), 28 pages.
https://doi.org/10.1145/3131671

1 INTRODUCTION

Mobile crowdsensing (MCS) has become a promising sensing paradigm for urban monitoring applications such as noise, air pollution, and traffic monitoring [15, 55, 60]. When an MCS task is conducted, *quality* and *budget* are two primary concerns of MCS organizers—while an MCS task requires high-quality sensed data that can well represent the whole target-sensing area, the organizer also aims to minimize the cost of recruiting participants and collecting data.

Existing work usually uses *coverage ratio*, that is, how many subareas (cells) of the target area have been covered, as a major data quality metric to measure whether sufficient sensed data has been collected from participants [2, 10, 18, 42, 49, 50, 56]. Since higher coverage ratio generally means better quality of the target-sensing map, previous research primarily takes full coverage [42, 50] or high probabilistic coverage as constraint [2, 18, 49, 56]. However, this means that an organizer has to collect at least one sensing value from each/most of the cells in the target area [49, 50]. As a consequence, data collection cost may still be high, especially when organizers carry out MCS campaigns at a large scale.

To further reduce sensing cost, one question arises: *is it possible to obtain a high-quality sensing map of the target area when only a small portion of the area is covered by the participants' contributed data*? To address this question, we first study the characteristics of the data collected in MCS tasks. We find that for a variety of sensing tasks, there are often certain data correlations in practice. In urban temperature or noise monitoring, for instance, there often exists a high spatiotemporal correlation [27, 34, 62]. In addition, different types of data may also have correlations with each other. For example, the humidity generally decreases when temperature increases [3], while PM2.5 and PM10 usually rise and drop together [28]. Such *intradata* (within the same type of data) and *interdata* (between different types of data) correlations render the high-accuracy data inference feasible, which, in turn, sheds light on the solution to achieving a high-quality sensing map from only sparsely sensed areas in MCS.

With these insights in mind, we propose to use *inference error*, rather than coverage ratio, to measure the data quality. By exploiting intradata and interdata correlations in data inference, we design an MCS task allocation framework, which aims at minimizing the number of the collected sensing values while ensuring that the inference error is lower than a predefined bound.

Actually, the proposed metric, inference error, is a more direct and practical quality measurement than coverage ratio. Although high coverage ratio generally means high quality, it is still an *indirect* quality measurement—even if an organizer has a particular data precision requirement for the task, there is no clear guideline for the organizer to decide how much coverage ratio is needed. In comparison, the organizer can easily set the inference error bound directly according to expectations on the data quality. Then, the organizer does not need to bother deciding how much coverage ratio is required; our task allocation framework would try to reduce the coverage ratio, that is, the data collection budget, while ensuring the organizer's data quality requirement is satisfied.

Despite the advantages, designing an inference-error-based task allocation framework would face the following challenges.

(1) *Data Inference:* How to exploit intradata and interdata correlations to infer missing data accurately? An inference algorithm is the core of our task allocation framework. The key to designing an effective inference algorithm is to efficiently incorporate various intradata and interdata correlations into the algorithm. Due to the sophisticated correlations among different kinds of data in reality, this is a nontrivial issue.

(2) *Quality Assessment:* How to quantitatively measure the inference error without knowing the true sensing values of unsensed cells? Accurately measuring the current inference error is also critical in our framework, as it decides whether the task allocation process can be stopped with a satisfactory data quality. If we stop too early, the quality requirement will not be reached; if we stop too late, we will sense more data than necessary. This is challenging, as we cannot compute the inference error directly by comparing the inferred value with the unknown ground truth.

(3) *Cell Selection:* Which cells should be chosen for sensing? To save the budget while ensuring the quality requirement for an MCS task, the organizer needs to select a minimum set of the cells for sensing. In order to choose this minimum cell combination, we need to identify the *salient* cells whose sensing values, if collected, can reduce the inference error to the desired extent. However, as we cannot know the true sensing value of a cell in advance, it is hard to predict how much that value can help decrease the inference error if collected.

With the aforementioned research objective and challenges, our main contributions are:

(1) We propose a novel practical MCS task allocation mechanism, called *SPACE-TA (SPArse and Cost-Effective Task Allocation)*, including three steps: *cell selection*, *data inference*, and *quality assessment*. These steps are seamlessly integrated to achieve cost-efficient task allocation with a data *inference* quality guarantee for the first time in MCS, as far as we know. Compared to our conference version [47] for the single-task scenario, the journal version adds the multitask scenario and enhances the single-task solution to more MCS applications, such as traffic monitoring.

(2) Effective methods are carefully designed for each step of SPACE-TA. As *quality assessment* has not been studied extensively to date, we propose three novel methods for three common types of inference errors, respectively: (1) Gaussian-distribution mean absolute error, (2) non-Gaussian-distribution mean absolute error, and (3) Bernoulli-distribution classification error with statistical analysis. To address the issues in *data inference* and *cell selection* stages, we adapt the techniques from *compressive sensing*, *transfer learning*, and *active learning* into both single-task and multitask MCS scenarios according to the real MCS applications' intradata and interdata correlations. With the three steps systematically integrated, SPACE-TA can iteratively select best cells for sensing, infer the unsensed data, and ensure that the inference quality meets a predefined requirement.

(3) We conduct extensive evaluations on real-life temperature [21], humidity [21], air quality [61], and traffic [40] monitoring datasets to verify the effectiveness of SPACE-TA. Specifically, leveraging the intradata correlations, for the temperature-monitoring task, SPACE-TA collects data from only 15.5% of the cells, on average, and can ensure the inferred mean absolute error below 0.25°C in 95% of the sensing cycles, while baseline approaches need to sense 18.0% to 26.5% more cells. When multiple MCS tasks run simultaneously, for example, for temperature and humidity sensing, by additionally considering interdata correlations, SPACE-TA further reduces the sensing cells by ∼10% compared to the single temperature/humidity-monitoring task.

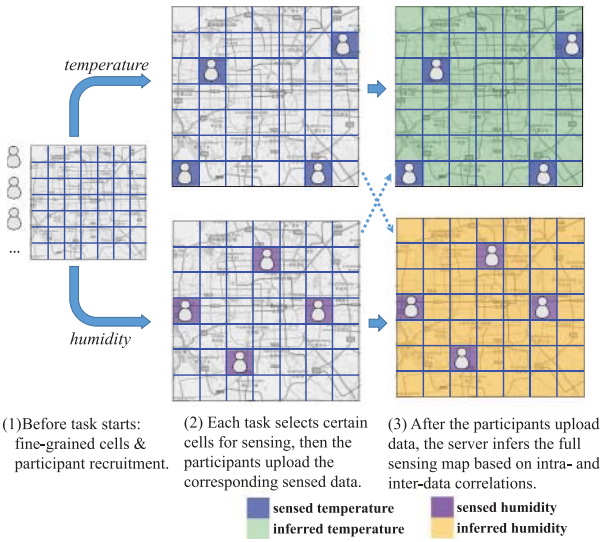


Fig. 1. Two MCS tasks, temperature and humidity monitoring, run at an urban area in one sensing cycle.

2 PROBLEM STATEMENT

In this section, we first illustrate a use case to motivate our research problem. We then define the key concepts, clarify the assumptions, and formulate the research problem.

2.1 Motivated Use Case

Figure 1 shows a use case to illustrate the basic process of our proposed framework: suppose that an MCS organizer launches two MCS tasks for environment monitoring: temperature and humidity. The target urban area has already been divided into cells according to the organizer's requirement. The organizer needs to update the full temperature/humidity-sensing map once every hour (sensing cycle); the data-quality requirement is that the mean absolute error for the whole area should be less than $0.25^{\circ}C$ (temperature) and 1.5% (humidity). To meet the data-quality requirement while minimizing the data-collection cost, the organizer actively selects a subset of the cells to sense temperature/humidity, where the sensed data is expected to reduce the inference error to the maximum extent. Based on the sensed temperature and humidity values, the temperature and humidity of the rest cells are inferred, exploiting both intradata and interdata correlations.

2.2 Definitions

With the previous use case in mind, we now define the key concepts used throughout this article.

Definition 1. Full Sensing Matrix. For an MCS task involving m cells and n sensing cycles, its *full sensing matrix* is denoted as F_{m*n}, where each entry $F[i, j]$ denotes the true sensed data of cell i in cycle j.

Definition 2. Cell-Selection Matrix. In a *cell-selection matrix* S_{m*n}, each entry $S[i, j]$ denotes whether or not the corresponding entry in the full sensing matrix $F[i, j]$ is selected for sensing: if cell i is selected for sensing in cycle j, $S[i, j] = 1$; otherwise, $S[i, j] = 0$.

Definition 3. Collected Sensing Matrix. A *collected sensing matrix* C_{m*n} records the actual collected data:

$$C = F \circ S,$$

where $\circ$ denotes the element-wise product of two matrices.

Definition 4. Sensing Matrix Inference Algorithm. A *sensing matrix inference algorithm* $\mathcal{R}$ attempts to reconstruct a *full sensing matrix* $\hat{F}_{m*n}$ from the *collected sensing matrix* C_{m*n}:

$$\mathcal{R}(C_{m*n}) = \hat{F}_{m*n} \approx F_{m*n}.$$

Definition 5. Inference Error. It quantifies the difference between the inferred sensing matrix $\hat{F}$ and the true sensing matrix F. In this article, we focus on the inference error of *each sensing cycle separately*. For sensing cycle k, the inference error is defined as:

$$\mathcal{E}_k = error(\hat{F}[:,k], F[:,k]),$$

where $F[:,k]$ is the kth column of F, that is, the true sensing values of all the m cells in cycle k, and $\hat{F}[:,k]$ contains the corresponding inferred sensing values by using the inference algorithm $\mathcal{R}$.

Note that the specific $error()$ function depends on the type of sensed data. In this article, we focus on two popular metrics: *mean absolute error* (for continuous values, e.g., temperature [4]), and *classification error* (for classification labels, e.g., air quality index (AQI) descriptors [61]).

Mean Absolute Error:

$$error(\hat{F}[:,k], F[:,k]) = \frac{\sum_{i=1}^{m} |\hat{F}[i,k] - F[i,k]|}{m} \tag{1}$$

Classification Error:

$$error(\hat{F}[:,k], F[:,k]) = 1 - \frac{\sum_{i=1}^{m} I(\psi(\hat{F}[i,k]), \psi(F[i,k]))}{m}, \tag{2}$$

where $\psi()$ is the function to map a value to its classification label; $I(x,y) = 1$ if $x = y$, otherwise 0.

Definition 6. (ϵ, p)*-quality.* For an MCS task lasting for n cycles, it satisfies (ϵ, p)*-quality*, iff

$$|\{k|\mathcal{E}_k \leq \epsilon, 1 \leq k \leq n\}| \geq n \cdot p,$$

where ϵ is a predefined inference error bound, and p is a predefined probability threshold to quantify the minimum fraction of the cycles whose errors should be lower than the bound ϵ.

Ideally, for a predefined error bound ϵ, we expect that an MCS task can keep the inference error lower than ϵ in *all* ($p = 1$) cycles. However, it is intractable for a real-life MCS task to satisfy $(\epsilon, 1)$-quality because we cannot know the accurate inference error $\mathcal{E}_k$ thus have to estimate it (as the ground truth F is not known in advance). Thus, we focus on the cases in which p is relatively high (e.g., 0.9 or 0.95) to guarantee the inference error bounded by ϵ in *most* cycles. This relaxation allows us to use the techniques from probability and statistics theory to tackle the problem, which will be illustrated later.

2.3 Assumptions

We make the following assumptions in this article.

Assumption 1. *Fixed Micropayment Incentive. For each task, a user gets a fixed amount of monetary incentive upon completion of one microtask allocated to that user and uploads the corresponding sensed data to the server.*

Assumption 1 means that the incentive is equal across different participants for each sample of sensed data of the same task. While it is a simple incentive mechanism, it is verified to be effective in many real-life MCS campaigns [36].

Assumption 2. *High-Quality Sensing. Every participant returns an accurate sensing value if a task is allocated to the participant.*

Assumption 2 also appears in other existing research work [52, 63]. We note that in real life, it is not always true due to possible issues such as sensor error or varying conditions. With an attractive incentive scheme in place, however, this assumption can be reasonable.

Assumption 3. *Not Moving Out During Sensing. After a participant receives a sensing task in a cell, the participant will not move out of the cell before completion of sensing.*

Assumption 3 ensures that if we allocate a sensing task to a participant in cell i, the participant's returned sensing value will actually represent cell i. This assumption can usually be satisfied if the sensing task does not consume much time. For example, with an embedded ambient temperature sensor, a smartphone can obtain the temperature reading in a few seconds;[1] for air-quality monitoring, usually the sensor needs 30s to 60s to be prepared to start sensing and then the sampling cycle is 2s to 10s [12, 19].

In summary, these assumptions are made for the following reasons:

- *Assumption 1* transforms our objective of minimizing data collection cost for each MCS task to minimizing the total number of selected sensing cells.
- Combining *assumptions 2 and 3*, for any MCS task t, we need to recruit only one participant in cell i during cycle j in order to get the true sensed data of task t from cell i in cycle j.

2.4 Problem Formulation

Based on the previous definitions and assumptions, next, we define our research problem if only one MCS task is conducted; then, we extend it to the multitask scenario.

Single-Task Scenario. We first formulate the research problem for the single-task scenario: **Given an MCS task with m cells and n cycles and a** *sensing matrix inference algorithm* $\mathcal{R}$**, we attempt to select a minimal subset of sensing cells during the whole sensing process (minimize the number of nonzero entries in the** *cell-selection matrix* S**), while ensuring that the** *inference errors* **of at least $n \cdot p$ cycles are below the predefined bound ϵ (satisfy (ϵ, p)-quality):**

$$\begin{aligned} \min \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} S[i,j] \\ s.t., \quad & |\{k|\mathcal{E}_k \le \epsilon, 1 \le k \le n\}| \ge n \cdot p \\ where \quad & \mathcal{E}_k = error(\hat{F}[:,k], F[:,k]) \\ & \hat{F} = \mathcal{R}(C),\ C = F \circ S \end{aligned}$$

Multitask Scenario. When an organizer launches multiple tasks simultaneously, it is natural to extend the above problem formulation of a single task to multiple tasks with the **objective of minimizing the number of selected cells for each task** (suppose totally z tasks; for each task

[1]The response time of the temperature/humidity sensor SHTC1 of Galaxy S4 is about 8s [1].

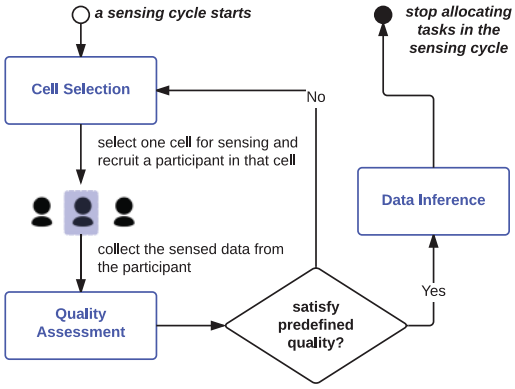


Fig. 2. Workflow of SPACE-TA for each MCS task in one cycle.

t, the quality requirement is (ϵ_t, p_t)-quality):

$$\begin{aligned} \min \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} S_t[i,j] \qquad t = 1...z \\ s.t., \quad & \forall t,\ |\{k|\mathcal{E}_{t,k} \le \epsilon_t, 1 \le k \le n\}| \ge n \cdot p_t \\ where \quad & \mathcal{E}_{t,k} = error(\hat{F}_t[:,k], F_t[:,k]) \\ & \hat{F}_t = \mathcal{R}_t(C_t),\ C_t = F_t \circ S_t \end{aligned}$$

Note that this is a multiobjective optimization problem. In this article, we try to optimize the task allocation process for each single task in parallel. The parallel mechanism is able to provide time-efficient task allocation, which is particularly important for large-scale MCS tasks.[2]

For both single- and multiple-task scenarios, as we cannot foresee the *full sensing matrix* F_t for any MCS task t, as it is impossible to obtain the optimal *cell selection matrix* S_t in reality. To overcome the difficulties, we propose *SPACE-TA*, which leverages an iterative process to select sensing cells in each cycle for each MCS task. Details are provided in the next section.

3 DESIGN OF SPACE-TA

In this section, we will elaborate on the algorithms used in the three stages of SPACE-TA: *data inference*, *quality assessment*, and *cell selection*. Before the detailed algorithm description, we first present an overview of the workflow of SPACE-TA to see the relationship among the three stages.

3.1 Overview

Figure 2 shows the workflow of the SPACE-TA for each running MCS task. In each cycle, SPACE-TA iteratively *selects the next salient cell for sensing* (**cell selection**) and waits for recruiting a participant present in that cell to get the sensed data, until the *estimated data quality* satisfies the predefined (ϵ, p)-*quality* requirement (**quality assessment**). Then, the task allocation stops and *missing data values of the unsensed cells are inferred* (**data inference**).

Figure 3 shows an example of the task allocation process of SPACE-TA in one sensing cycle for one task. Suppose that the target-sensing area contains five cells and the fifth sensing cycle starts

[2]The parallel solution may have some limitations in minimizing the total costs of different tasks, especially when the tasks have significantly different costs. We will discuss this in Section 6.3.

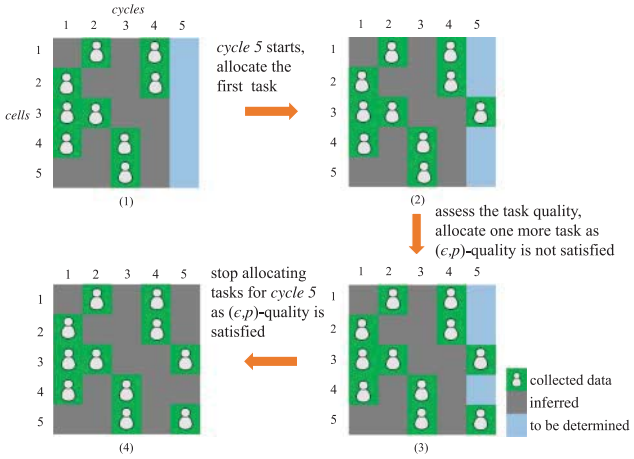


Fig. 3. A running example of SPACE-TA.

currently. In the beginning, no sensing data is collected in *cycle 5* (Figure 3-1). SPACE-TA works as follows:

(1) SPACE-TA selects the first salient cell (*cell 3*) and allocates a sensing task to one participant appearing in *cell 3* (cell selection algorithm is elaborated in Section 3.4). This participant performs the sensing task and returns the sensing data (Figure 3-2).
(2) Then, given the sensing data already collected, SPACE-TA decides if the data quality satisfies the predefined (ϵ,p)-quality requirement (the quality assessment algorithm is described in Section 3.3). If the data quality does not meet the quality requirement, SPACE-TA selects the next cell for sensing (*cell 5* in Figure 3-3). In this way, SPACE-TA continues allocating tasks to new cells and collects sensing data, until the data quality of already collected sensing data satisfies the quality requirement.
(3) Given the collected sensing values, SPACE-TA infers the sensing values of remaining unselected cells (Figure 3-4; data inference algorithm is illustrated in Section 3.2).

Next, we elaborate the three stages in SPACE-TA, respectively.

3.2 Data Inference

To infer the full sensing matrix from the partially collected sensing values, *Compressive Sensing* (*CS*) is commonly used in the literature [27, 63]. In this section, we first introduce the basic idea of CS, and then illustrate an enhanced version of CS, called *Spatiotemporal Compressive Sensing* (*STCS*), which considers the spatial and temporal correlations among the environmental data explicitly to further improve the inference performance [27, 58]. We use STCS to infer missing values in SPACE-TA, given its improved inference accuracy over normal CS and the other methods [27].

3.2.1 CS: Compressive Sensing. Given a partially collected sensing matrix C, *compressive sensing* infers the full sensing matrix $\hat{F}$ based on the low-rank property:

$$\begin{aligned} &\min\ rank(\hat{F}) \\ &s.t.,\ \hat{F} \circ S = C \end{aligned} \tag{3}$$

Directly solving this problem is hard because it is nonconvex. Based on the *singular value decomposition*, that is, $\hat{F} = LR^T$, and compressive sensing theory [6, 13, 35], existing works have theoretically proved that minimizing the rank of $\hat{F}$ is equivalent to minimizing the sum of L and R's Frobenius norms under certain conditions [58]:

$$\begin{aligned} &\min\ ||L||_F^2 + ||R||_F^2 \\ &s.t.,\ LR^T \circ S = C. \end{aligned} \tag{4}$$

In practice, while real-life collected data often contain some noises, the above optimization problem is usually converted as follows [27, 58, 63]:

$$\min\ \lambda(||L||_F^2 + ||R||_F^2) + ||LR^T \circ S - C||_F^2, \tag{5}$$

where λ is used to make a trade-off between rank minimization and accuracy fitness. To get the optimal $\hat{F}$, we use an alternating least squares [27, 58, 63] procedure to estimate L and R iteratively ($\hat{F} = LR^T$).

3.2.2 STCS: Spatiotemporal Compressive Sensing. As environment data such as temperature usually exhibits strong spatial and temporal correlations, explicit spatiotemporal correlations have been introduced into compressive sensing in recent work [27, 58], called *spatiotemporal compressive sensing*, which focuses on the optimization function below:

$$\begin{aligned} \min\ &\lambda_r(||L||_F^2 + ||R||_F^2) + ||LR^T \circ S - C||_F^2 \\ &+ \lambda_s||\mathbb{S}(LR^T)||_F^2 + \lambda_t||(LR^T)\mathbb{T}^T||_F^2, \end{aligned} \tag{6}$$

where $\mathbb{S}$ and $\mathbb{T}$ are spatial and temporal constraint matrices, respectively; λ_r, λ_s, and λ_t are chosen to balance the weights of different elements in the optimization problem.

Similar to the *CS* optimization problem in Equation (5), the above *STCS* optimization problem in Equation (6) could be solved by using alternating least squares [27, 58]. We elaborate below on our strategies of choosing the temporal and spatial constraint matrices.

Temporal constraint matrix ($\mathbb{T}$): As in [27, 58], we choose the temporal constraint matrix $\mathbb{T}$ as $Toeplitz(0, 1, -1)_{n*n}$ (total n sensing cycles), which considers the temporal correlation in the following manner—for a specific cell, its sensing values in two continuous sensing cycles should be similar.

Spatial constraint matrix ($\mathbb{S}$): The spatial constraint matrix $\mathbb{S}$ is used to express the correlations between the sensed data from different cells. How to construct this matrix is dependent on specific MCS tasks. In this article, we apply two commonly used strategies in the literature [27, 58]. For environment monitoring, we use physical distance [27] to model the correlation between cell i and j, $c_{i,j}$, as $1/distance(cell_i, cell_j)$; for traffic monitoring, we conduct a correlation analysis on historical traffic data [58] to model $c_{i,j}$. Finally, we get the spatial constraint matrix as

$$\mathbb{S}_{i,i} = -1;\ \mathbb{S}_{i,j} = c_{i,j} / \sum_{k \neq i} c_{i,k},\ \text{if } i \neq j.$$

3.2.3 CoSTCS: Collective STCS. When an MCS organizer conducts multiple crowdsensing tasks to collect different types of data simultaneously, it is possible that the inference performance of one type of data can be boosted by considering the collected data of another type, because different types of data—for example, temperature/humidity [3], PM2.5/PM10 [28]—may present some inherent correlations. Thus, we propose a method called *collective spatiotemporal compressive sensing (CoSTCS)* to simultaneously infer the missing values for different types of data considering such interdata correlations. Specifically, CoSTCS is inspired by the *collective matrix factorization* technique [43] in the *transfer learning* research area [31]. Collective matrix factorization supposes that

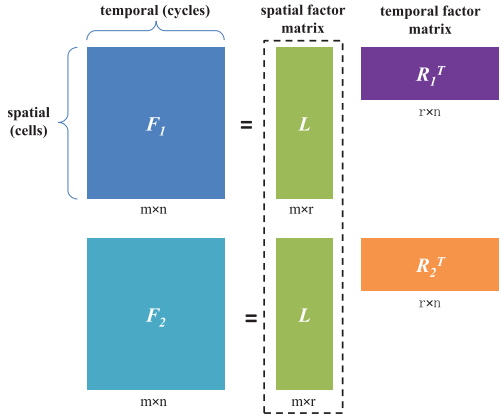


Fig. 4. Illustration of collective matrix factorization (assume the spatial factor matrix L is the shared factor).

during the matrix decomposition process, different matrices share one particular factor matrix that encodes the common dynamics from multiple types of data. Recall that in STCS, we decompose the inferred sensing matrix $\hat{F}$ into two factor matrices L and R. Then, in CoSTCS, for each MCS task t, its sensing matrix $\hat{F}_t$ is decomposed into L_t and R_t, and we assume that one of the two factor matrices is the same for all the tasks' sensing matrices. Without loss of generality, here we assume that for all the tasks, L_t is the same, that is, $\forall t,\ L_t = L$. Inferring the missing values of k MCS tasks is formalized as the following optimization problem:

$$\begin{aligned}\min \sum_{t=1}^{k} (\lambda_r(||L||_F^2 + ||R_t||_F^2) + ||LR_t^T \circ S_t - C_t||_F^2 \\ + \lambda_s||\mathbb{S}(LR_t^T)||_F^2 + \lambda_t||(LR_t^T)\mathbb{T}^T||_F^2).\end{aligned} \tag{7}$$

With traditional collective matrix factorization, we have to select only one from L and R to fix. As shown in Figure 4, L is the spatial factor matrix and R is the temporal factor matrix; with only one of them fixed, we can consider only one type of interdata correlation. However, in reality, sometimes both spatial and temporal interdata correlations exist for multitasks. Then, *how can we incorporate both interdata correlations into inference?*

To address this issue, we solve the problem in Equation (7) twice, each time fixing L or R, and thus obtain two inferred sensing matrices for each task t, denoted as $\hat{F}_{t,L}$ and $\hat{F}_{t,R}$, respectively. Then, we use a weighted averaging method to aggregate the two inferred matrices to exploit both spatial and temporal interdata correlations in multitask data inference:

$$\hat{F}_t = w\hat{F}_{t,L} + (1-w)\hat{F}_{t,R}, \tag{8}$$

where w can be set according to the extent of spatial and temporal interdata correlations observed in real applications.

3.3 Quality Assessment

In SPACE-TA, for each sensing cycle, assessing the inference error and accordingly deciding when to stop task allocation is critical. If we stop too early, the server might not collect enough data to achieve the predefined (ϵ, p)-quality; if we stop too late, the server might collect redundant data, which would lead to the waste of the organizer's budget. In this article, we focus on two widely

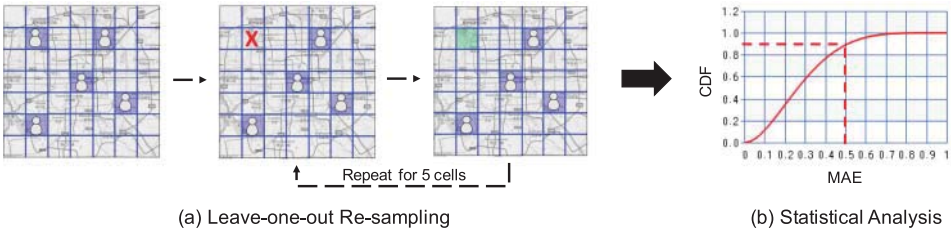


Fig. 5. An illustrative example of LOO-SA method.

used error metrics, *mean absolute error* (MAE; see Equation (1)) and *classification error* (CE; see Equation (2)).

We propose the *Leave-One-Out Statistical-Analysis* (*LOO-SA*) method to assess inference error and decide the stopping criterion for each sensing cycle. First, LOO-SA uses the *leave-one-out* resampling method to obtain a set of reinferred sensing data with the corresponding true collected data. Then, comparing the reinferred data to the true collected data, *Bayesian* or *Bootstrap* analysis is leveraged to assess whether the current data quality can satisfy the predefined (ϵ, p)-quality requirement.

3.3.1 Leave-One-Out Resampling. In statistics, *leave-one-out* is a popular resampling method to measure the performance for many prediction and classification algorithms [23]. Suppose that we have m true observations, the basic idea of leave-one-out is that for each time, we leave one observation out and using the other $m-1$ observations (as training data set) to make a prediction for the excluded observation. By running this process on all m observations, we get m predictions accompanying the m true observations, which can be used to estimate the prediction error.

To run the leave-one-out resampling, in each iteration, LOO-SA temporarily removes one piece of collected data of the current cycle k and then runs the inference algorithm $\mathcal{R}$ to reinfer the removed data. After enumerating all the collected data in cycle k, we finally get two vectors $\mathbf{x}$ and $\mathbf{y}$: $\mathbf{x}$ stores the true collected data for the current cycle k, while $\mathbf{y}$ stores the corresponding reinferred data by using leave-one-out. Suppose that we have already collected data from m' cells, then:

$$\mathbf{x} = \langle x_1, x_2, \ldots, x_{m'} \rangle, \quad \mathbf{y} = \langle y_1, y_2, \ldots, y_{m'} \rangle$$

where x_i is the ith ground truth data collected in cycle k, and y_i is the corresponding reinferred data by leaving x_i out of the collected data.

Figure 5(a) illustrates an example of the leave-one-out method when the data of five cells have been collected. Then, for each cell c_i with collected data, we use the data collected from the other four cells and the data collected in previous cycles (not shown in the figure) to infer the data of c_i. We repeat this for five cells and get five inferred data $\langle y_1, y_2, \ldots, y_5 \rangle$ and their corresponding ground truth collected data $\langle x_1, x_2, \ldots, x_5 \rangle$. Based on the ground truth set $\mathbf{x}$ and the leave-one-out reinferred set $\mathbf{y}$, the next section will discuss how to assess whether (ϵ, p)-quality is satisfied.

3.3.2 Statistical Analysis for Quality Estimation. To assess whether (ϵ, p)-quality is satisfied, based on the leave-one-out $\mathbf{x}$ and $\mathbf{y}$, we conduct a statistical analysis to estimate the probability distribution of the inference error $\mathcal{E}$ (MAE or CE) for the current cycle. We use an example to demonstrate why the inference error probability distribution can help quality assessment. Figure 5(b) shows the cumulative probability distribution of the estimated MAE for a temperature MCS task. We can see that the probability of the inference error $\mathcal{E}_k \leq 0.5°C$ is larger than 0.9, which is expected to satisfy $(0.5°C, 0.9)$-quality.

With this example in mind, the problem of assessing whether a task can satisfy (ϵ, p)-quality can be converted to calculate the probability of $\mathcal{E}_k \leq \epsilon$, that is, $P(\mathcal{E}_k \leq \epsilon)$, for the current cycle k. If $P(\mathcal{E}_k \leq \epsilon) > p$ can hold for every cycle k, then (ϵ, p)-quality is expected to be satisfied. Next, we propose leveraging two statistical analysis methods for estimating $P(\mathcal{E}_k \leq \epsilon)$: *Bayesian* [16] and *Bootstrap* [14] analysis. Bayesian analysis can address the scenario in which the error metric is *normal-distributed MAE* and *CE*, while Bootstrap analysis can address the scenario in which *MAE does not follow normal distribution*. In SPACE-TA, if the estimated $P(\mathcal{E}_k \leq \epsilon)$ is larger than p, we stop the task allocation for the current cycle; otherwise, we continue selecting more cells for collecting data (Section 3.4).

(1) Bayesian Analysis. In Bayesian analysis, we see $\mathcal{E}_k$ as an unknown parameter with a *prior* probability distribution $g(\mathcal{E}_k)$.[3] Based on our observation $\boldsymbol{\theta}$ (obtained from the leave-one-out reinferred data, which will be explained later), we update the probability distribution of $\mathcal{E}_k$, getting the *posterior* probability distribution $g(\mathcal{E}_k|\boldsymbol{\theta})$ according to the *Bayes's Theorem*:

$$g(\mathcal{E}_k|\boldsymbol{\theta}) = \frac{f(\boldsymbol{\theta}|\mathcal{E}_k)g(\mathcal{E}_k)}{\int_{-\infty}^{\infty} f(\boldsymbol{\theta}|\mathcal{E}_k)g(\mathcal{E}_k)d\mathcal{E}_k,} \tag{9}$$

where $f(\theta|\mathcal{E}_k)$ is the *likelihood* function that represents the conditional probability of observing $\boldsymbol{\theta}$ given $\mathcal{E}_k$.

The posterior $g(\mathcal{E}_k|\boldsymbol{\theta})$ is thus the estimated probability distribution of $\mathcal{E}_k$, based on which we can approximate $P(\mathcal{E}_k \leq \epsilon)$:

$$P(\mathcal{E}_k \leq \epsilon) \approx \int_{-\infty}^{\epsilon} g(\mathcal{E}_k|\boldsymbol{\theta})d\mathcal{E}_k. \tag{10}$$

If $P(\mathcal{E}_k \leq \epsilon) \geq p$, then SPACE-TA stops the task allocation for the current cycle k and waits for the start of the next cycle; otherwise, SPACE-TA continues selecting a new cell to collect sensing data.

Next, we describe how to compute the posterior $g(\mathcal{E}_k|\theta)$ for two widely used error metrics, MAE (for continuous value, e.g., temperature [4]) and CE (for classification label, e.g., AQI descriptor [61]).

Bayesian Analysis for Mean Absolute Error. When $\mathcal{E}_k$ is defined as MAE (Equation (1)), we use the absolute difference of $\mathbf{y}$ (leave-one-out reinferred data) and $\mathbf{x}$ (true collected data) as the observation $\boldsymbol{\theta}$ (suppose that m' sensing values have been collected in the current cycle):

$$\boldsymbol{\theta} = \langle \theta_1, \theta_2, \ldots, \theta_{m'} \rangle = abs(\mathbf{y} - \mathbf{x}) \tag{11}$$

$$= \langle |y_1 - x_1|, |y_2 - x_2|, \ldots, |y_{m'} - x_{m'}| \rangle. \tag{12}$$

After inspecting our evaluation temperature dataset (which will be described in detail later in the evaluation), we find that the MAE in each sensing cycle follows the normal distribution. Figure 6 shows the histogram of the standardized MAE (i.e., MAE divided by the standard deviation of MAE in each cycle) when 10% of the cells are sensed and the remaining 90% are inferred. Thus, by assuming that the sampled absolute errors satisfy the normal distribution around mean $\mathcal{E}_k$ and variance σ^2, we get the likelihood function:

$$f(\boldsymbol{\theta}|\mathcal{E}_k) : \ \theta_i = |y_i - x_i| \sim \mathcal{N}(\mathcal{E}_k, \sigma^2).$$

[3]The prior distribution is often selected as a noninformative probability distribution (such as uniform distribution) if we do not have specific prior knowledge about $\mathcal{E}_k$.

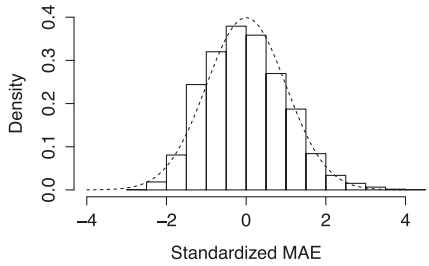


Fig. 6. Histogram of mean absolute error with fitted normal curve (temperature, following the normal distribution).

Calculating the posterior $g(\mathcal{E}_k|\boldsymbol{\theta})$ from the above likelihood function and observation is a classic Bayesian statistics problem: *inferring normal mean with unknown variance*, which can be solved by fixing the variance σ^2 to the sample variance s^2 and then directly calculating the posterior $g(\mathcal{E}_k|\boldsymbol{\theta})$ by t-distribution [5]. For the prior $g(\mathcal{E}_k)$, we select the *Jeffreys' flat prior* [24]: $g(\mathcal{E}_k) = 1, \quad \forall \mathcal{E}_k$. Then, the posterior $g(\mathcal{E}_k|\boldsymbol{\theta})$ satisfies the following (m'-1) degree-of-freedom t-distribution:

$$g(\mathcal{E}_k|\boldsymbol{\theta}) \sim t_{m'-1}(\bar{\theta}, s^2), \tag{13}$$

where $\bar{\theta}$ is the sample mean of the values in $\boldsymbol{\theta}$.

Bayesian Analysis for Classification Error. We now show how we use Bayesian analysis to estimate the posterior distribution for CE $\mathcal{E}_k$ (Equation (2)). First, as our data inference algorithm $\mathcal{R}$ deals with continuous values, we map $\mathbf{x}$ and $\mathbf{y}$ to their corresponding classification labels using the mapping function $\psi()$ in Equation (2), for example, for the PM2.5 AQI value between 0 and 50, we map it into the AQI descriptor label "Good." Then, we use the $I()$ function on $\psi(\mathbf{x})$ and $\psi(\mathbf{y})$ to get our observation θ:

$$\begin{aligned}\theta &= \langle \theta_1, \theta_2, \ldots, \theta_{m'} \rangle = I(\psi(\mathbf{x}), \psi(\mathbf{y})) \\ &= \langle I(\psi(x_1), \psi(y_1)), I(\psi(x_2), \psi(y_2)), \ldots, I(\psi(x_{m'}), \psi(y_{m'})) \rangle.\end{aligned}$$

$I(\psi(x_i), \psi(y_i))$ is either 1 (success, $\psi(x_i) = \psi(y_i)$) or 0 (failure, $\psi(x_i) \neq \psi(y_i)$), and $\mathcal{E}_k$ is exactly the failure ratio. Suppose that each θ_i is independent, then it satisfies the *Bernoulli* distribution with the probability of $1 - \mathcal{E}_k$:

$$f(\theta|\mathcal{E}_k): \ \theta_i = I(\psi(x_i), \psi(y_i)) \sim Bernoulli(1 - \mathcal{E}_k).$$

Then, the problem to infer the posterior $g(\mathcal{E}_k|\boldsymbol{\theta})$ is converted to a classic Bayesian statistics problem, *Coin Flipping* [5, 16]. We choose the *uniform prior* for $\mathcal{E}_k$: $g(\mathcal{E}_k) = 1$ for $0 \leq \mathcal{E}_k \leq 1$. Then, the posterior for $\mathcal{E}_k$ follows *Beta* distribution [5, 16]:

$$g(\mathcal{E}_k|\boldsymbol{\theta}) \sim Beta(m' - z + 1, z + 1) \tag{14}$$

where $z = \sum_{i=1}^{m'} \theta_i$, that is, the number of successes.

(2) Bootstrap Analysis. While Bayesian analysis can deal with the normal-distributed MAE, the MAE of some MCS tasks may not follow the normal distribution well. For example, in the traffic speed monitoring dataset (elaborated later in the evaluation), the MAE of inferred traffic speed of unsensed roads (suppose that 90% of roads are unsensed) is seriously right skewed, as shown in Figure 7. For such a scenario, the previous Bayesian analysis cannot work, as the normal-distribution assumption does not hold.

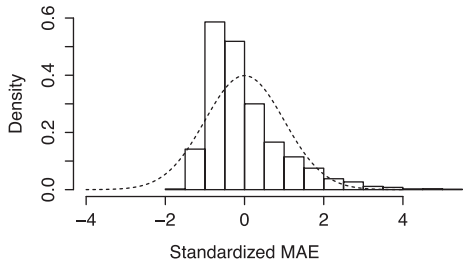


Fig. 7. Histogram of mean absolute error with fitted normal curve (traffic speed, not following the normal distribution).

To address this issue, we propose using Bootstrap to estimate $P(\mathcal{E}_k \leq \epsilon)$. Similar to Bayesian analysis, Bootstrap [14] is another widely used method to infer the unknown statistic of a *population* in which the *observations* are sampled. The advantage of Bootstrap is that no assumption on the distribution of the observations needs to be made. The basic idea of Bootstrap is to construct a number (usually several thousand) of resamples from the observations *with replacement*, and the size of each resample is equal to the original observations. Then, the unknown statistic of the population, e.g., mean, can be inferred from the bootstrapping resamples. Note that in Bootstrap, to get a good estimation of $P(\mathcal{E}_k \leq \epsilon)$, the size of the original observations cannot be too small. According to [9], a reasonable size of the observations should be larger than ten.[4]

Specifically, to estimate the MAE for an MCS task such as traffic-speed monitoring, we first obtain the observations $\boldsymbol{\theta}$ in the same way as for Bayesian analysis (Equation (11)). Then, we construct n bootstrapping resamples $\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \ldots \boldsymbol{\theta}_n^*$, by resampling θ with replacement, and $size(\boldsymbol{\theta}_i^*) = size(\boldsymbol{\theta})$. A direct method to estimate the probability of $P(\mathcal{E}_k \leq \epsilon)$ is to see how many resamples' means are not larger than ϵ, that is, $|\{i|mean(\boldsymbol{\theta}_i^*) \leq \epsilon\}|/n$, which is called the *percentile confidence interval* of Bootstrap. While the percentile confidence interval shows a rough idea of Bootstrap analysis, a more theoretically sound method, called *bias-corrected and accelerated Bootstrap (BCa)*, is able to reduce the bias in the percentile confidence interval and boost the convergence speed (i.e., using smaller n to a convergent result). In SPACE-TA, for time efficiency, we adopt a state-of-the-art method to approximate BCa analytically, called *approximation bootstrap confidence interval (ABC)*. ABC uses Taylor Series Expansions to approximate the Bootstrap resampling results thus avoiding the large number of Bootstrap replications. This approximation requires that the estimated statistic $\mu(\theta)$ is defined smoothly in θ; fortunately, the "mean" considered in MAE estimation satisfies this requirement. Interested readers can refer to Chapters 14 and 22 in [14] for more details about BCa and ABC.

Computation Complexity of LOO-SA. As there are two phases for the computation of LOO-SA, we discuss the computation complexity of both phases. First, to use leave-one-out to estimate the sensing error, LOO-SA needs to run the inference algorithm $\mathcal{R}$ for m' times, where m' is the number of the already collected sensing values in the current cycle. This time consumption dominates the runtime so that the computation complexity is $O(m' \cdot T_{\mathcal{R}})$ for the leave-one-out part, where $T_{\mathcal{R}}$ is the complexity of the inference algorithm $\mathcal{R}$. For the second part of Bayesian analysis, recalling Equation (13) and Equation (14), we can simply use two distributions, t-distribution and

[4]In our evaluation, when using Bootstrap, we set the minimum number of sensing cells in each cycle to 10 and then start quality assessment. This performs well on traffic monitoring. For other MAE scenarios (e.g., temperature), usually less than 10 sensed values per cycle are needed to ensure the quality and their MAE follows the normal distribution well. We thus adopt Bayesian analysis for them.

Beta distribution respectively, to calculate the posterior for MAE and CE, which is much faster than the leave-one-out part; for the ABC method used in Bootstrap analysis, it also runs pretty fast as it is an analytic method, not needing Bootstrap resampling for several thousand times. In summary, the computation complexity of LOO-SA is dominated by the leave-one-out part, which is $O(m' \cdot T_{\mathcal{R}})$. If m' is large, sequentially executing LOO-SA might consume much time. Fortunately, though we need to run $\mathcal{R}$ for m' times, each run is independent; thus, LOO-SA can be easily parallelized to accelerate as needed.

3.4 Cell Selection

When the estimated data quality of LOO-SA does not satisfy the predefined requirement, SPACE-TA will continue selecting more cells where at least one participant is present for sensing. During this process, selecting some salient cells for sensing may reduce the overall sensing error more significantly; for example, the missing values of some cells might incur more inference errors and are thus more uncertain. If SPACE-TA can identify these salient cells, the number of the allocated tasks can possibly be reduced to make the data quality satisfy the predefined (ϵ, p)-quality requirement earlier, compared to other simple cell selection methods such as random selection.

Based on recent research advances in *active learning* on matrix completion, we use a method proposed in [8], called *Query by Committee (QBC)*, to select the salient cell to allocate the next task (*committee* here refers to a set of various data inference algorithms). QBC attempts to use each algorithm in the committee to infer the full sensing matrix. Then, it allocates the next task to the cell with the largest variance among the inferred values of different algorithms [8]. If the cell with the largest variance has no participants, then the second largest is selected, and so forth.

In SPACE-TA, the committee includes *CS*, *STCS*, *CoSTCS* (for the multitask scenario), *KNN-S*, and *KNN-T*. *CS*, *STCS*, and *CoSTCS* have been described previously, and *KNN-S* and *KNN-T* use the classic *K-Nearest Neighbors (KNN)* [11] method to interpolate missing values. For a missing value, KNN uses a weighted average of the values of the k nearest neighbors. In sensing matrix inference, we can perform KNN on columns or rows, that is, using spatial (KNN-S) or temporal (KNN-T) K nearest neighbors. Specifically, for a missing value $F[i, j]$ (cell i in cycle j), KNN-S attempts to find K nearest spatial neighbors $F[i', j]$ (weight $\propto 1/distance(cell_i, cell_{i'})$), while KNN-T attempts to find K nearest temporal neighbors $F[i, j']$ (weight $\propto 1/|j - j'|$).

In the multitask scenario, for each type of sensed data, we select the next salient cell individually. In SPACE-TA, considering the runtime efficiency, the selection processes for different tasks are conducted in parallel (i.e., cell selection is sequential within one task, while parallel between different tasks). It is worth noting that, in terms of the data quality, this parallel selection method may not be as good as the sequential method, in which we select the {task, cell} pair alternatively (e.g., in temperature–humidity monitoring, $\{tem, cell_{1,t}\} \rightarrow \{hum, cell_{1,h}\} \rightarrow \{tem, cell_{2,t}\} \rightarrow \{hum, cell_{2,h}\} \rightarrow \cdots$). The reason is that, using the parallel method, the data collected for individual tasks in the same iteration is obtained at the same time and thus cannot help each other in data inference. In contrast, as the sequential method alternately gets the sensed data, it can immediately leverage the data collected for one task to improve the inference for other tasks. However, the sequential method is not practically scalable with a relatively large number of tasks running simultaneously.[5] Our future work will explore a more effective and scalable way of cell

[5]Later in the evaluation, we will see that the runtime of SPACE-TA to select one cell is roughly 10s. Then, by assuming the time for a participant to return data as 10s, SPACE-TA can get data from ~180 cells/h sequentially. While this number is probably enough for one MCS task, it does not seem sufficient for a moderate number of simultaneous tasks. For example, when 10 tasks run simultaneously, each task can only get data from 18 cells/h.

Table 1. Statistics of Three Evaluation Datasets

	SensorScope	U-Air	TaxiSpeed
City	Lausanne (Switzerland)	Beijing (China)	Beijing (China)
Data	temperature, humidity	PM2.5, PM10	traffic speed
Cell size	50m*30m	1km*1km	one road segment
Number of cells	57	36	100
Cycle length	0.5h	1h	1h
Duration	7d	11d	4d
Error metric	mean absolute error	classification error	mean absolute error
Mean ± Std.	6.04 ± 1.87 °C (temperature) 84.52 ± 6.32 % (humidity)	79.11 ± 81.21 (PM2.5) 63.12 ± 48.56 (PM10)	13.01 ± 6.97 m/s

selection in the multitask scenario, especially referring to the recent advances in the interdisciplinary research area of active learning and transfer learning, such as in [59].

Computation Complexity of QBC. The runtime of the QBC method is primarily spent on using all the algorithms in the committee to infer the sensing matrix. Suppose that, for each inference algorithm $\mathcal{R}_i$ in the committee, the computation complexity is $T_{\mathcal{R}_i}$; then the complexity of QBC is $O(\sum_i T_{\mathcal{R}_i})$. If the committee contains more algorithms, then, running QBC sequentially will cost more time. Like LOO-SA, since the executions of different inference algorithms are independent, QBC can also be parallelized to improve runtime performance.

4 EVALUATION

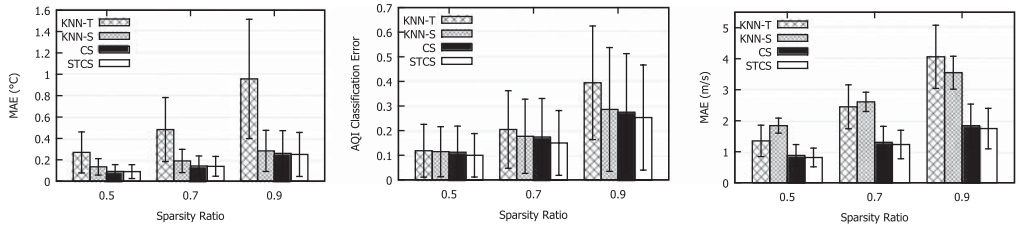
In this section, based on three real-life sensing datasets, we evaluate SPACE-TA on various MCS applications, including temperature, humidity, air quality, and traffic monitoring. The evaluation is carried out following two steps. First, we verify that a single MCS task can run well under the SPACE-TA framework by considering intradata correlations. Second, we demonstrate that if several correlated MCS tasks run simultaneously, SPACE-TA can further reduce the number of sensed cells by considering the interdata correlations.

4.1 Experiment Setup

To evaluate the real-world applicability of our work, we use three real-life sensing datasets, *SensorScope* [21], *U-Air* [61], and *TaxiSpeed* [40], which include various types of sensed data in representative MCS applications, such as temperature, air quality, and traffic speed. While the *SensorScope* and *U-Air* datasets are collected by sensor networks and static stations, respectively, the MCS participants can also obtain the data using smartphones (as in [12, 19]). The summary statistics of the three datasets are listed in Table 1.

SensorScope [21]: The *SensorScope* dataset contains various environment readings, for example, temperature and humidity, from a sensor network deployed in the EPFL campus (500m × 300m). For our evaluation, we divide the target area into 100 cells (each cell is 50m × 30m), and find that 57 cells are deployed with valid sensors. We use the *MAE* (Equation (1)) to evaluate the quality of the environment readings in this dataset.

U-Air [61]: The *U-Air* dataset consists of PM2.5, PM10, and NO2 AQI values reported by 36 air-quality monitoring stations in Beijing. For our evaluation, as in [61], we split the Beijing urban area into 1km × 1km cells and use only the cells where the stations are situated. To measure the data quality for AQI values, we follow the methods used in [61]—each AQI value is classified to a range called a *descriptor*, which is used as the basis for computing the *CE* (Equation (2)). Six



(a) Temperature (b) PM2.5 (c) Traffic Speed

Fig. 8. Inference error (with standard deviation) on three single-task scenarios.

Table 2. Fraction of the Cycles Whose Errors are Lower than the Error Bound ϵ

	Temperature		**PM2.5**		**Traffic Speed**	
ϵ	$0.25°C$	$0.30°C$	$6/36$	$9/36$	$2.0m/s$	$2.5m/s$
$p = 0.90$	0.915	0.919	0.904	0.912	0.895	0.895
$p = 0.95$	0.943	0.949	0.944	0.965	0.987	0.953

levels of descriptors are defined: *Good* (0–50), *Moderate* (51–100), *Unhealthy for Sensitive Groups* (101–150), *Unhealthy* (150–200), *Very Unhealthy* (201–300), and *Hazardous* (>300).

TaxiSpeed [40]: The *TaxiSpeed* dataset includes the speed information for road segments in Beijing for 4 days (September 12 through 15, 2013) based on 33,000 taxis' GPS trajectories. According to [63], we refer to each road segment as a cell, and select a target area that has 100 road segments with valid speed values. We use the *MAE* as the metric for speed inference.

4.2 Single-Task Scenario

In the evaluation of the single-task scenario, we focus on three individual MCS applications from three datasets, respectively: *temperature* (*SensorScope*), *PM2.5* (*U-Air*), and *traffic speed* (*TaxiSpeed*).

4.2.1 Inference Error. First, we aim to verify the effectiveness of *STCS* in inferring missing values for temperature, PM 2.5, and traffic speed compared to the other state-of-the-art matrix inference algorithms described before, including CS, KNN-S, and KNN-T. The parameters of STCS and CS are selected as in [27].

Figure 8 shows the inference error (with standard deviation) of different algorithms on temperature, PM2.5, and traffic-monitoring scenarios, respectively. In this experiment, we iteratively consider each sensing cycle k as the latest cycle, infer the full sensing matrix based on the collected sensing matrix from cycle *1* to k, and calculate the inference error for the cycle k. The x axis, that is, *sparsity ratio*, denotes the fraction of unsensed entries in the collected sensing matrix. Similar to the literature [27, 58], our evaluation results also show the improved accuracy of STCS over the other methods, verifying that compressive sensing is effective in inferring the missing data, such as temperature, air quality, and traffic speed, especially when the explicit spatiotemporal correlations are incorporated.

4.2.2 Quality Requirement Satisfaction. Then, we evaluate the effectiveness of the quality assessment algorithm *LOO-SA* to see whether it can satisfy the predefined quality requirement. We use various settings of (ϵ, p)-quality to see what fraction of sensing cycles can actually keep the inference error less than ϵ. Table 2 shows the results for three single-task scenarios. For p, we purposely set it to a large value as 0.90 and 0.95, that is, ensuring *most* (90% or 95%) sensing

Table 3. Fraction of the Cycles Whose Errors are Lower than $0.25^{\circ}C$ (Temperature), 9/36 (PM2.5) or $2m/s$ (Traffic Speed) for Different Approaches

	SPACE-TA	RAND-TA	FIX-TA-k
Temperature			
$p = 0.9$	0.915	0.900	0.911 ($k = 9$)
$p = 0.95$	0.943	0.943	0.949 ($k = 12$)
PM2.5			
$p = 0.9$	0.912	0.916	0.884 ($k = 16$)
$p = 0.95$	0.965	0.969	0.961 ($k = 18$)
Traffic Speed			
$p = 0.9$	0.895	0.908	0.908 ($k = 24$)
$p = 0.95$	0.987	0.974	0.960 ($k = 28$)

cycles' error to be less than the predefined error bound ϵ, which we think is a more reasonable and realistic scenario than small p for MCS organizers. For ϵ, we vary it from 0.25°C to 0.30°C for the temperature, 6/36 to 9/36 for the PM2.5, and $2m/s$ to $2.5m/s$ for the traffic speed, respectively. Note that for PM25, the error bound $X/36$ represents that to satisfy this error bound, more than $36 - X$ cells have the correct AQI level.

From Table 2, we see that for any predefined error bound ϵ, the actual fraction of the cycles whose errors are less than ϵ is quite similar to the p predefined in (ϵ, p)-quality. Even though the actual fractions sometimes are slightly less than the predefined p due to the intrinsic probabilistic characteristics of our algorithm, the values are still quite near the predefined p (in our settings, the largest gap between the predefined p and the actual fraction is only $0.007 = 0.95 - 0.943$). Based on these results, we verify that, by using LOO-SA as the quality assessment algorithm, SPACE-TA can adequately satisfy the predefined (ϵ, p)-quality.

4.2.3 Number of Sensed Cells. After selecting the best inference algorithm and verifying the effectiveness of the quality assessment algorithm, now we focus on analyzing the research objective—*how many sensed cells could SPACE-TA reduce while ensuring a certain data quality?*

To compare with SPACE-TA, we use the following baselines:

- **FIX-TA-*k*** fixes the selected cell number k in each sensing cycle while still using QBC to actively select cells for sensing. Compared to FIX-TA-k, SPACE-TA shows the benefit brought by LOO-SA, which helps the organizer decide when to stop the task allocation, thus adaptively adjusting the number of the sensed cells for different cycles.
- **RAND-TA** randomly selects the next cell for sensing, but still uses LOO-SA as the data quality assessment method. Compared to RAND-TA, SPACE-TA shows the advantage of applying QBC to select the salient cells for sensing.

In the temperature-monitoring scenario, for the predefined (ϵ, p)-quality, we set the error bound ϵ as 0.25°C and p as 0.9 or 0.95. Before comparing the number of sensed cells, we need to ensure that all methods can achieve a similar task quality. While SPACE-TA has already been verified to be able to satisfy (ϵ, p)-quality in the previous section, now we need to check the two baselines. Table 3 shows the results. We can see that RAND-TA can also satisfy (ϵ, p)-quality well, as it adopts LOO-SA to assess quality like SPACE-TA. For FIX-TA-k, we tune k to achieve a similar task quality, which leads to $k = 9$ for $p = 0.9$ and $k = 12$ for $p = 0.95$.

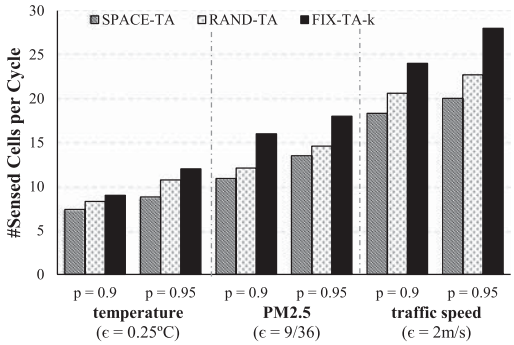
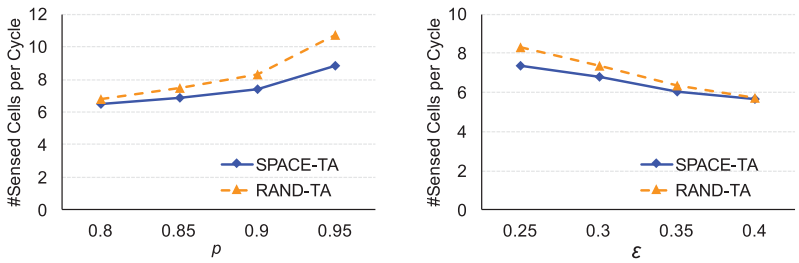


Fig. 9. Number of sensed cells on three single-task scenarios.



(a) Vary p ($\epsilon = 0.25°C$) (b) Vary ϵ ($p = 0.9$)

Fig. 10. Number of sensed cells in temperature sensing with different quality requirements.

As all methods can satisfy similar task quality, we compare their numbers of selected cells in Figure 9 (leftmost part). When $p = 0.9$, SPACE-TA can allocate 11.1% fewer tasks than RAND-TA and 18.0% fewer tasks than FIX-TA-9; when $p = 0.95$, SPACE-TA outperforms RAND-TA and FIX-TA-12 by assigning 18.0% and 26.5% fewer tasks, respectively. Specifically, SPACE-TA allocates tasks to only 12.9% (15.5%) cells, on average, while ensuring that the inference error is below 0.25°C in 90% (95%) of the cycles. To further study how the change of (ϵ, p)-quality will impact the evaluation results, we conduct more experiments on temperature sensing, as shown in Figure 10, by varying p and ϵ. Generally, with a higher quality requirement (i.e., larger p or smaller ϵ), SPACE-TA can gain better performance improvement over RAND-TA. This is probably due to the fact that when the data quality requirement is high, our cell selection strategy has more space (i.e., more data is needed) to optimize the task allocation for reducing the sensing costs.

For the other two single-task scenarios, we get similar observations (Table 3 and Figure 9). For the PM2.5 and traffic monitoring, SPACE-TA allocates 7.5% to 31.9% and 11.9% to 28.5% fewer tasks than the baseline methods, respectively, under the same data quality.

In the previous experiments, we assume that there always is at least one participant in the selected cell. Here, we investigate how the performance of SPACE-TA will change if some cells do not have participants. Figure 11 shows the number of sensed cells in temperature sensing when some selected cells may not have any participants. If probability of no-participant cells is 0.1, it means that in one cycle, each cell in the target area has 10% probability without a participant. Generally, with the increase of this probability, the average number of sensed cells slightly increases. Even with the increase, the required cell number remains at a low level (smaller than RAND-TA

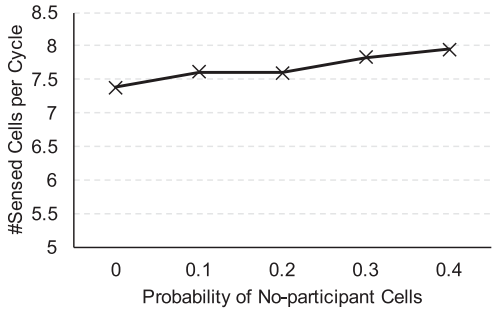
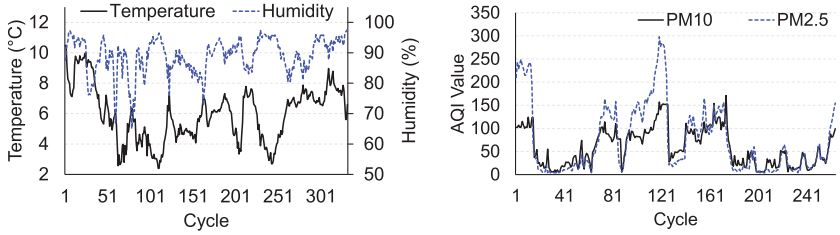


Fig. 11. Number of sensed cells in temperature sensing if some cells do not have participants ($\epsilon = 0.25°C$, $p = 0.9$).



(a) Temperature-Humidity (b) PM2.5-PM10

Fig. 12. Interdata correlations in multitask scenarios.

Table 4. Interdata Pearson Correlation P-values
(Value Significant at the Level of 0.05 is Bold)

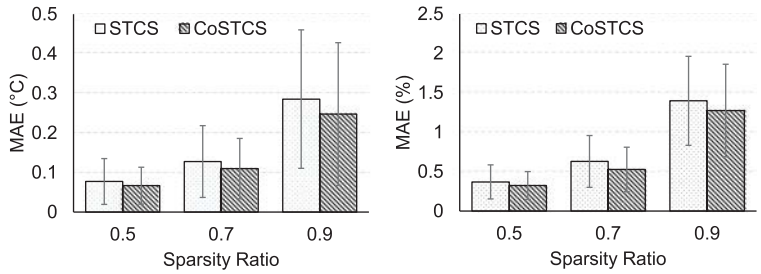
	P-Values	
	Temporal	*Spatial*
Temperature–Humidity	**0.009**	0.233
PM2.5-PM10	**0.000**	**0.013**

when every cell always has participants), meaning that SPACE-TA is robust to the condition when certain cells do not have any participants.

4.3 Multitask Scenario

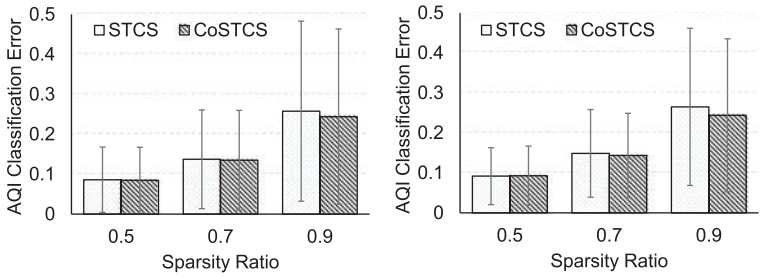
In multitask scenarios, we use *SensorScope* and *U-Air* datasets, as they contain multiple types of sensed data. Specifically, we focus on two multitask cases: *Temperature–Humidity (SensorScope)* and *PM2.5–PM10 (U-Air)*.

4.3.1 Inference Error. Existing literature has pointed out that a negative correlation exists between temperature and relative humidity [3], while a positive correlation appears between PM2.5 and PM10 [28]. An empirical study on our evaluation dataset also complies with the literature, as shown in Figure 12. A significance test of the Pearson correlation shows which type of interdata correlation (spatial or temporal) dominates in our applications: for temperature and humidity, the temporal interdata correlation is generally significant while the spatial one is not; for PM2.5 and PM10, both spatial and temporal interdata correlations are significant (Table 4). To this end, for the temperature–humidity scenario, we only keep the temporal collective matrix factorization



(a) Temperature (b) Humidity

Fig. 13. Inference error (with standard deviation) of STCS and CoSTCS for the temperature–humidity scenario.



(a) PM2.5 (b) PM10

Fig. 14. Inference error (with standard deviation) of STCS and CoSTCS for the PM2.5–PM10 scenario.

result in CoSTCS ($w = 0$ in Equation (8)); for the PM2.5–PM10 scenario, we keep both spatial and temporal collective matrix factorization results in CoSTCS ($w = 0.5$ in Equation (8)).

To incorporate the *negative* correlation between temperature and humidity into CoSTCS, we build the humidity-sensing matrix using the complementary value of the original humidity $h\%$, that is, $(100 - h)\%$. To normalize temperature to the same scale as humidity (0–1) when running CoSTCS, we divide the original temperature value by 12, as the temperature values range from 0 to 12°C in the evaluation dataset.

Figure 13 plots the inference error in the temperature–humidity scenario using STCS and CoSTCS, which shows that by considering the interdata correlations, CoSTCS can further reduce the inference error by around 10% to 15% for both temperature and humidity under different sparsity settings. Similarly, CoSTCS also outperforms STCS in the PM2.5 to PM10 scenario, as shown in Figure 14. These results verify that the interdata correlations exploited in CoSTCS can boost the inference accuracy.

Note that to get such inference performance improvement, the weight in Equation (8) needs to be carefully tuned according to the real-life spatial or temporal interdata correlation. Figure 15 shows for the temperature–humidity case how weight w impacts the inference accuracy. As mentioned previously, the interdata correlations in the temperature–humidity scenario is dominated by the temporal one. We verify that the inference error is actually the smallest when w is set to 0 (only considering temporal interdata correlation), and gradually increases when increasing w. Setting w to 1, that is, considering only the spatial interdata correlation, results in the highest inference error (even larger than STCS without considering any interdata correlations).

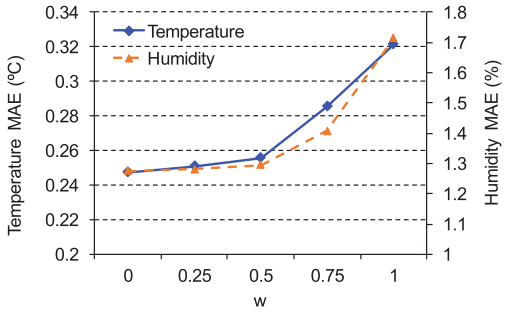


Fig. 15. Inference error for the temperature–humidity scenario with different w in Equation (8) (sparsity is 0.9; $w = 0$ refers to only temporal interdata correlation, and $w = 1$ refers to only spatial interdata correlation).

Table 5. Number of Sensed Cells for SPACE-TA with STCS and CoSTCS

	Temperature–Humidity				**PM2.5–PM10**			
	temperature ($\epsilon = 0.25°C$)		**humidity** ($\epsilon = 1.5\%$)		**PM2.5** ($\epsilon = 9/36$)		**PM10** ($\epsilon = 9/36$)	
	STCS	***CoSTCS***	*STCS*	***CoSTCS***	*STCS*	***CoSTCS***	*STCS*	***CoSTCS***
$p = 0.9$	7.4 (0.915)	**6.9** (0.891)	7.6 (0.923)	**7.0** (0.899)	10.9 (0.912)	**10.5** (0.898)	12.5 (0.894)	**11.7** (0.902)
$p = 0.95$	8.8 (0.943)	**8.0** (0.946)	8.8 (0.942)	**7.6** (0.942)	13.5 (0.965)	**12.4** (0.949)	13.7 (0.979)	**12.6** (0.943)

Note: The results of CoSTCS are in bold and the values in brackets are the actual fraction of sensing cycles whose errors are below the bound ϵ.

Table 6. Runtime for Each Stage of SPACE-TA

	Temperature	**PM2.5**	**Traffic Speed**
Data Inference	0.45s	0.39s	0.71s
Quality Assessment	4.43s	4.75s	7.99s
Cell Selection	1.04s	0.91s	1.29s

4.3.2 Number of Sensed Cells. As CoSTCS is able to improve the inference accuracy in multitask scenarios, we expect that the total number of sensed cells of SPACE-TA can also be decreased by leveraging CoSTCS while the data quality is still guaranteed.

Table 5 shows the number of the sensed cells for the multitask scenarios. In the temperature–humidity scenario, we set the error bound ϵ to 0.25°C for temperature and 1.5% for humidity; in the PM2.5 to PM10 scenario, we set ϵ to 9/36 for both tasks. From Table 5, we first see that the actual fraction of the cycles whose inference errors below ϵ is close to the predefined p, verifying that the (ϵ, p)-quality is well satisfied. Then, we can also observe that the number of sensed cells of SPACE-TA with CoSTCS is consistently smaller than that with STCS. Specifically, in the temperature–humidity scenario, the number of sensed cells is reduced by 5% to 13%; while in the PM2.5 to PM10 scenario, the reduction is 4% to 8%.

4.4 Runtime Analysis

Finally, we study the runtime of SPACE-TA. As SPACE-TA can inherently deal with multiple tasks in parallel on different CPUs, our performance evaluation is conducted assuming that one CPU (computer) runs one task. We run the experiments on a laptop (Intel Core i7-3612QM, 8GB RAM, Windows 7) with Python 2.7. Table 6 shows the runtime for different parts of SPACE-TA. The most time-consuming part is the quality assessment step, which needs ~8s, on average, in

traffic-speed monitoring. As described previously, the quality assessment method, LOO-SA, is suitable for being parallelized, which can help improve its performance if more than one CPU can be used for one task. In summary, on our experimental setup, SPACE-TA spends ~10s to do one allocation iteration, that is, estimating the inference quality once and, if it cannot meet the predefined (ϵ, p)-quality, finds the next sensing cell. Thus, for the data requiring a few seconds to sense, for example, temperature, if we can find a participant and receive the participant's data in 10s, SPACE-TA can collect data from ~180 cells in an hour; even for the air quality sensing that needs 60s to get a valid reading [12, 19], SPACE-TA can allocate tasks to ~50 cells in an hour. We believe that this efficiency can meet most real-life MCS scenarios, especially with more powerful servers in a SPACE-TA deployment environment and more efficient smartphone-equipped sensors in the future. Furthermore, if an MCS task does need faster runtime speed, we can accelerate SPACE-TA by letting it select more than one salient cell for sensing in each iteration. This may incur redundant collected data, however, which could increase the budget. Our future work will study this trade-off between runtime speed and budget saving in more detail.

5 RELATED WORK

We review the related work from four perspectives: (1) task allocation in MCS, (2) compressive sensing applications, (3) active learning for matrix completion, and (4) transfer learning for matrix completion. Finally, we discuss the differences of this article from the previous conference version of this work [47].

5.1 MCS Task Allocation Mechanisms

Existing work about the task allocation for MCS applications mainly uses the coverage ratio of the target area as the major quality metric. In early work on this topic, Reddy et al. [37] attempted to recruit a predefined number of participants to maximize spatial coverage. Later, various work attempts to extend this type of coverage-maximization participant recruitment to different considerations, for example, participants' reverse auction incentive mechanism [22] and travel time budget [20]. On the other hand, work such as [2, 18, 42, 50, 51, 56] attempts to minimize the incentive budget and/or energy consumption under a full or high probabilistic spatial and/or temporal coverage constraint. Recently, the coverage-based task allocation studies are also extended to multiple MCS task scenarios [17, 30, 45, 46]. Compared to the existing work, we do not use the coverage ratio as the quality metric. Instead, we use a more essential metric, that is, inference error, to represent data quality, based on which we attempt to reduce the number of sensed cells to help the MCS organizers lower the budget.

In this article, our incentive model adopts a micropayment style, which is verified effective in real MCS campaigns [36]. Many MCS studies also design more complicated incentive models used in task allocation, such as auction based [29] and game-theory based [53]. Interested readers can find a comprehensive study of existing MCS incentive mechanisms in [57].

5.2 Compressive Sensing Applications

Compressive sensing theory [6, 13, 35] is increasingly becoming a powerful tool to reconstruct a sparse vector or matrix based on the sparsity property of vector or low rank property of matrix. Thus, a large number of applications based on compressive sensing have appeared, such as network traffic reconstruction [58], environmental data recovery [27, 33, 34], road traffic monitoring [63], and face recognition on smartphones [41]. While the above work primarily focuses on designing the effective algorithms to minimize the reconstruction error under different scenarios, our objective is to minimize the number of the allocated tasks and meet a predefined data quality requirement, thus opening up the possibility of using any suitable inference techniques. Indeed,

in SPACE-TA, the compressive sensing algorithms proposed in the above work, for example, *STCS* [27, 58], is just one possible implementation for inferring the missing values of the unsensed cells. Recently, assuming that a fixed number of data need to be collected in each cycle and different sensing data require different costs, Xu et al. [52] study the trade-off between total costs and overall data quality in compressive sensing. In contrast to [52], instead of fixing the number of data collected in each cycle, we aim to minimize the number of data collected in each cycle while still ensuring the data quality.

Besides compressive sensing, there are still other state-of-the-art methods to infer missing values for unsensed cells, such as multichannel singular spectrum analysis [26] and expectation maximization [38]. As existing work [27, 63] shows that compressive sensing outperforms these methods, we currently do not explore them in SPACE-TA.

5.3 Active Learning for Matrix Completion

The key idea behind active learning is to make a machine-learning algorithm achieve higher accuracy with less training data, if it is allowed to select the training data from which the algorithm learns [39]. To solve the problem of choosing the best cells for sensing, the recent techniques on active learning for matrix completion [8, 25, 44], which employ different criteria to actively choose the entry in a matrix, are all applicable. Currently, we use *QBC* in SPACE-TA due to its easy implementation and good performance, which has been shown in [8].

5.4 Transfer Learning for Matrix Completion

For the multitask MCS scenario, different types of sensed data may have certain inter-correlations that can facilitate the missing data inference. To leverage such correlations to improve inference accuracy, state-of-the-art techniques in transfer learning [31] for matrix completion are potentially useful. As a representative technique in transfer learning, *collective matrix factorization* [43] is powerful tool for inferring missing values for multiple matrices by jointly factorizing several matrices with the constraints of sharing certain latent features. The idea has been widely used and extended to various applications, such as link prediction [7], item recommendation [32], and music source separation [54], especially when there are data sources in heterogeneous domains. Like these works, we adopt collective matrix factorization to improve inference accuracy in the multitask scenario.

5.5 Difference from Our Previous Work

Compared with the previous conference version [47], this article has made two distinct improvements. First, to increase task allocation efficiency when multiple MCS tasks run simultaneously, based on matrix cofactorization [43], we propose a *collective spatiotemporal compressive sensing* method to boost inference accuracy by considering the interdata correlations between different tasks, and evaluate its efficiency on two multitask scenarios, temperature–humidity and PM2.5 to PM10. Second, to apply SPACE-TA to an MCS task whose MAE of inferred sensing values do not follow normal distribution, we design a new data quality assessment method based on *Bootstrap* [9, 14], and verify its effectiveness on the MCS task of traffic-speed monitoring. Furthermore, this article is also a detailed technical contribution to the research direction of *sparse mobile crowdsensing*, which was proposed in our perspective paper [48].

6 DISCUSSIONS

6.1 Cell Size Configuration

In our evaluation, the setting of the cell size follows existing literature for specific applications such as air quality [61] and traffic-speed [63] monitoring. Generally, the smaller the cell size, the

higher the sensing map precision can be achieved, but it will also incur higher costs. Therefore, the selection of cell size depends primarily on the MCS organizers' quality needs and budget. In addition, *adaptive* cell size configuration may be an interesting research direction, where the cell size can be set differently across the whole sensing area. The basic idea is that for the area where the sensing data values vary significantly, the cell size could be set a bit smaller, and vice-versa.

6.2 Different Incentives for the Same Task

In this work, we assume that for one task, the incentive costs for collecting each sample of data are always the same, regardless of where and who contributes the data. In practice, the incentive mechanism could be more complex, as the data from different cells may not cost the same. For example, the costs may be inversely proportional to the number of participants in that cell [30], or the network signal strength of the cell [52]. In such a case, the cell selection method needs to be revised to take the diverse cell costs into consideration.

6.3 Further Cost Optimization Opportunities in Multitask Scenario

Our current solution tries to select the sensing cells for multiple tasks in parallel. This mechanism has good runtime efficiency but, due to its parallelism, it has certain limitations in minimizing total costs when different tasks have significantly different costs. Consider a scenario that sensing PM2.5 costs \$1 while PM10 costs \$10. Our current solution selects PM2.5 and PM10 simultaneously and may result in collecting data from 5 cells for both PM2.5 and PM10, leading to \$55. However, with the interdata correlations, we may be able to reduce the costs by sensing more PM2.5 and less PM10 with the same quality guarantee, for example, 10 cells for PM2.5 and 3 cells for PM10, leading to \$40. Our future work may consider how to design a more cost-efficient task allocation strategy in such a scenario.

7 CONCLUSION

In this article, while ensuring a certain data quality, we attempt to reduce the number of the sensing cells in MCS tasks by considering both intradata and interdata correlations. To that end, we propose a novel task allocation framework, *SPACE-TA*, combining the state-of-the-art compressive sensing, statistical analysis, active learning, and transfer learning techniques to select a small set of sensing cells in each cycle while inferring the missing values of the remaining cells and ensuring the inference error below a predefined bound. Evaluation results on real-world temperature, humidity, air-quality and traffic-monitoring datasets show the effectiveness and applicability of SPACE-TA.

REFERENCES

[1] SENSIRION 2017. SHTC1 - Digital Temperature and Humidity Sensor. Retrieved Sept 29, 2017 from https://www.sensirion.com/en/environmental-sensors/humidity-ensors/digital-humidity-sensor-for-consumer-electronics-and-iot/.

[2] Asaad Ahmed, Keiichi Yasumoto, Yukiko Yamauchi, and Minoru Ito. 2011. Distance and time based node selection for probabilistic coverage in people-centric sensing. In *SECON*. 134–142.

[3] Richard G. Allen, Luis S. Pereira, Dirk Raes, Martin Smith, and others. 1998. Crop evapotranspiration-guidelines for computing crop water requirements-FAO irrigation and drainage paper 56. *FAO, Rome* 300, 9, D05109.

[4] Paul V. Bolstad, Lloyd Swift, Fred Collins, and Jacques Régnière. 1998. Measured and predicted air temperatures at basin to regional scales in the southern Appalachian mountains. *Agricultural and Forest Meteorology* 91, 3, 161–176.

[5] William M. Bolstad. 2007. *Introduction to Bayesian statistics*. John Wiley & Sons, Hoboken, NJ.

[6] Emmanuel J. Candès and Benjamin Recht. 2009. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics* 9, 6, 717–772.

[7] Bin Cao, Nathan N. Liu, and Qiang Yang. 2010. Transfer learning for collective link prediction in multiple heterogenous domains. In *Proceedings of ICML*. 159–166.

[8] Shayok Chakraborty, Jiayu Zhou, Vineeth Balasubramanian, Sethuraman Panchanathan, Ian Davidson, and Jieping Ye. 2013. Active matrix completion. In *ICDM*. 81–90.

[9] Michael R. Chernick. 2011. *Bootstrap Methods: A Guide for Practitioners and Researchers*. Vol. 619. John Wiley & Sons, Hoboken, NJ.

[10] Yohan Chon, Nicholas D. Lane, Yunjong Kim, Feng Zhao, and Hojung Cha. 2013. Understanding the coverage and scalability of place-centric crowdsensing. In *UbiComp*. 3–12.

[11] Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13, 1, 21–27.

[12] Srinivas Devarakonda, Parveen Sevusu, Hongzhang Liu, Ruilin Liu, Liviu Iftode, and Badri Nath. 2013. Real-time air quality monitoring through mobile sensing in metropolitan areas. In *UrbComp*. 15:1–15:8.

[13] David L. Donoho. 2006. Compressed sensing. *IEEE Transactions on Information Theory* 52, 4, 1289–1306.

[14] Bradley Efron and Robert J. Tibshirani. 1994. *An Introduction to the Bootstrap*. CRC Press, Boca Raton, FL.

[15] Raghu K. Ganti, Fan Ye, and Hui Lei. 2011. Mobile crowdsensing: Current state and future challenges. *IEEE Communications Magazine* 49, 11, 32–39.

[16] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. 2013. *Bayesian Data Analysis*. CRC Press, Boca Raton, FL.

[17] Bin Guo, Yan Liu, Wenle Wu, Zhiwen Yu, and Qi Han. 2017. Activecrowd: A framework for optimized multitask allocation in mobile crowdsensing systems. *IEEE Transactions on Human-Machine Systems* 47, 3, 392–403.

[18] Sara Hachem, Animesh Pathak, and Valérie Issarny. 2013. Probabilistic registration for large-scale mobile participatory sensing. In *PerCom*. 132–140.

[19] David Hasenfratz, Olga Saukh, Silvan Sturzenegger, and Lothar Thiele. 2012. Participatory air pollution monitoring using smartphones. In *2nd International Workshop on Mobile Sensing*.

[20] Shibo He, Dong-Hoon Shin, Junshan Zhang, and Jiming Chen. 2014. Toward optimal allocation of location dependent tasks in crowdsensing. In *INFOCOM*. 745–753.

[21] François Ingelrest, Guillermo Barrenetxea, Gunnar Schaefer, Martin Vetterli, Olivier Couach, and Marc Parlange. 2010. SensorScope: Application-specific sensor network for environmental monitoring. *ACM Transactions on Sensor Networks* 6, 2, 17:1–17:32.

[22] Luis Gabriel Jaimes, Idalides Vergara-Laurens, and Miguel A. Labrador. 2012. A location-based incentive mechanism for participatory sensing systems with budget constraints. In *PerCom*. 103–108.

[23] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Springer, New York.

[24] Harold Jeffreys. 1946. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 186, 1007, 453–461.

[25] Rasoul Karimi, Christoph Freudenthaler, Alexandros Nanopoulos, and Lars Schmidt-Thieme. 2011. Non-myopic active learning for recommender systems based on matrix factorization. In *IRI*. 299–303.

[26] Dmitri Kondrashov and Michael Ghil. 2006. Spatio-temporal filling of missing points in geophysical data sets. *Nonlinear Processes in Geophysics* 13, 2, 151–159.

[27] Linghe Kong, Mingyuan Xia, Xiao-Yang Liu, Guangshuo Chen, Yu Gu, Min-You Wu, and Xue Liu. 2014. Data loss and reconstruction in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems* 25, 11, 2818–2828.

[28] Rakesh Kumar and Abba Elizabeth Joseph. 2006. Air pollution concentrations of PM2. 5, PM10 and NO2 at ambient and kerbsite and their correlation in Metro city–Mumbai. *Environmental Monitoring and Assessment* 119, 1–3, 191–199.

[29] Juong-Sik Lee and Baik Hoh. 2010. Dynamic pricing incentive for participatory sensing. *Pervasive and Mobile Computing* 6, 6, 693–708.

[30] Yan Liu, Bin Guo, Yang Wang, Wenle Wu, Zhiwen Yu, and Daqing Zhang. 2016. TaskMe: Multi-task allocation in mobile crowd sensing. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 403–414.

[31] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10, 1345–1359.

[32] Weike Pan, Nathan N. Liu, Evan W. Xiang, and Qiang Yang. 2011. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *Proceedings of IJCAI*.

[33] Giorgio Quer, Riccardo Masiero, Gianluigi Pillonetto, Michele Rossi, and Michele Zorzi. 2012. Sensing, compression, and recovery for wsns: Sparse signal modeling and monitoring framework. *IEEE Transactions on Wireless Communications* 11, 10, 3447–3461.

[34] Rajib Kumar Rana, Chun Tung Chou, Salil S. Kanhere, Nirupama Bulusu, and Wen Hu. 2010. Ear-phone: An end-to-end participatory urban noise mapping system. In *IPSN*. 105–116.
[35] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. 2010. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review* 52, 3, 471–501.
[36] Sasank Reddy, Deborah Estrin, Mark Hansen, and Mani Srivastava. 2010. Examining micro-payments for participatory sensing data collections. In *UbiComp*. ACM, 33–36.
[37] Sasank Reddy, Deborah Estrin, and Mani Srivastava. 2010. Recruitment framework for participatory sensing data collections. In *Pervasive*. 138–155.
[38] Tapio Schneider. 2001. Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values. *Journal of Climate* 14, 5, 853–871.
[39] Burr Settles. 2010. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin–Madison.
[40] Jingbo Shang, Yu Zheng, Wenzhu Tong, Eric Chang, and Yong Yu. 2014. Inferring gas consumption and pollution emission of vehicles throughout a city. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1027–1036.
[41] Yiran Shen, Wen Hu, Mingrui Yang, Bo Wei, Simon Lucey, and Chun Tung Chou. 2014. Face recognition on smartphones via optimised sparse representation classification. In *IPSN*. 237–248.
[42] Xiang Sheng, Jian Tang, and Weiyi Zhang. 2012. Energy-efficient collaborative sensing with mobile phones. In *INFOCOM*. 1916–1924.
[43] Ajit P. Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. In *KDD*. ACM, 650–658.
[44] Dougal J. Sutherland, Barnabás Póczos, and Jeff Schneider. 2013. Active learning and search on low-rank matrices. In *KDD*. 212–220.
[45] Jiangtao Wang, Yasha Wang, Daqing Zhang, Feng Wang, Yuanduo He, and Liantao Ma. 2017. PSAllocator: Multi-task allocation for participatory sensing with sensing capability constraints. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM, 1139–1151.
[46] Jiangtao Wang, Yasha Wang, Daqing Zhang, Leye Wang, Haoyi Xiong, Abdelsalam Helal, Yuanduo He, and Feng Wang. 2016. Fine-grained multitask allocation for participatory sensing with a shared budget. *IEEE Internet of Things Journal* 3, 6, 1395–1405.
[47] Leye Wang, Daqing Zhang, Animesh Pathak, Chao Chen, Haoyi Xiong, Dingqi Yang, and Yasha Wang. 2015. CCS-TA: Quality-guaranteed online task allocation in compressive crowdsensing. In *UbiComp*. ACM, 683–694.
[48] Leye Wang, Daqing Zhang, Yasha Wang, Chao Chen, Xiao Han, and Abdallah M'hamed. 2016. Sparse mobile crowdsensing: Challenges and opportunities. *IEEE Communications Magazine* 54, 7, 161–167.
[49] Haoyi Xiong, Daqing Zhang, Guanling Chen, Leye Wang, Vincent Gauthier, and Laura Barnes. 2016. iCrowd: Near-optimal task allocation for piggyback crowdsensing. *IEEE Transactions on Mobile Computing* 15, 2010–2022.
[50] Haoyi Xiong, Daqing Zhang, Leye Wang, and Hakima Chaouchi. 2015. EMC3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint. *IEEE Transactions on Mobile Computing* 14, 7, 1355–1368.
[51] Haoyi Xiong, Daqing Zhang, Leye Wang, J. Paul Gibson, and Jie Zhu. 2015. EEMC: Enabling energy-efficient mobile crowdsensing with anonymous participants. *ACM Transactions on Intelligent Systems and Technology* 6, 3, 39.
[52] Liwen Xu, Xiaohong Hao, Nicholas D. Lane, Xin Liu, and Thomas Moscibroda. 2015. Cost-aware compressive sensing for networked sensing systems. In *IPSN*. 130–141.
[53] Dejun Yang, Guoliang Xue, Xi Fang, and Jian Tang. 2012. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*. ACM, 173–184.
[54] Jiho Yoo, Minje Kim, Kyeongok Kang, and Seungjin Choi. 2010. Nonnegative matrix partial co-factorization for drum source separation. In *Proceedings of ICASSP*. IEEE, 1942–1945.
[55] Daqing Zhang, Leye Wang, Haoyi Xiong, and Bin Guo. 2014. 4W1H in mobile crowd sensing. *IEEE Communications Magazine* 52, 8, 42–48.
[56] Daqing Zhang, Haoyi Xiong, Leye Wang, and Guanlin Chen. 2014. CrowdRecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint. In *UbiComp*. 703–714.
[57] Xinglin Zhang, Zheng Yang, Wei Sun, Yunhao Liu, Shaohua Tang, Kai Xing, and Xufei Mao. 2016. Incentives for mobile crowd sensing: A survey. *IEEE Communications Surveys & Tutorials* 18, 1, 54–67.
[58] Yin Zhang, Matthew Roughan, Walter Willinger, and Lili Qiu. 2009. Spatio-temporal compressive sensing and internet traffic matrices. In *SIGCOMM*. 267–278.
[59] Zihan Zhang, Xiaoming Jin, Lianghao Li, Guiguang Ding, and Qiang Yang. 2016. Multi-domain active learning for recommendation. In *Proceedings of AAAI*.

[60] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology* 5, 3, 38.
[61] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. 2013. U-Air: When urban air quality inference meets big data. In *KDD*. 1436–1444.
[62] Yu Zheng, Tong Liu, Yilun Wang, Yanmin Zhu, Yanchi Liu, and Eric Chang. 2014. Diagnosing New York City's noises with ubiquitous data. In *UbiComp*. 715–725.
[63] Yanmin Zhu, Zhi Li, Hongzi Zhu, Minglu Li, and Qian Zhang. 2013. A compressive sensing approach to urban traffic estimation with probe vehicles. *IEEE Transactions on Mobile Computing* 12, 11, 2289–2302.