



This is an electronic reprint of the original article. This reprint may differ from the original in pagination and typographic detail.

Gao, Hui; Feng, Jianhao; Xiao, Yu; Zhang, Bo; Wang, Wendong A UAV-assisted Multi-task Allocation Method for Mobile Crowd Sensing

Published in: IEEE Transactions on Mobile Computing

DOI: 10.1109/TMC.2022.3147871

Published: 01/01/2022

Document Version Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version: Gao, H., Feng, J., Xiao, Y., Zhang, B., & Wang, W. (2022). A UAV-assisted Multi-task Allocation Method for Mobile Crowd Sensing. *IEEE Transactions on Mobile Computing*, *22*(7). https://doi.org/10.1109/TMC.2022.3147871

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

A UAV-assisted Multi-task Allocation Method for Mobile Crowd Sensing

Hui Gao, Jianhao Feng, Yu Xiao, *Member, IEEE*, Bo Zhang, *Member, IEEE*, and Wendong Wang, *Member, IEEE*

Abstract—Mobile crowd sensing (MCS) with human participants has been proposed as an efficient way of collecting data for smart cities applications. However, there often exist situations where humans are not able or reluctant to reach the target areas, due to for example traffic jams or bad road conditions. One solution is to complement manual data collection with autonomous data collection using unmanned aerial vehicles (UAVs) equipped with various sensors. In this paper, we focus on the scenarios of UAV-assisted MCS and propose a task allocation method, called "UMA" (UAV-assisted Multi-task Allocation method) to optimize the sensing coverage and data quality. The method incentivizes human participants to contribute sensing data from nearby points of interest (Pols), with a limited budget. Meanwhile, the method jointly considers the optimization of task assignment and trajectory scheduling. It schedules the trajectories of UAVs, considering the locations of human participants, other UAVs and Pols which are rarely visited by human participants. In detail, UAVs take care of two tasks in our proposal. One is to calibrate the data collected by the human participants whom the UAVs come across along their trajectories. The other is to collect data from the Pols which are not covered by other UAVs or human participants. We apply deep reinforcement learning to schedule UAVs moving trajectories and sensing activities in order to minimize the overall energy cost. We evaluate the proposed scheme via simulation using two real data sets. The results show that our proposal outperforms the compared methods, in terms of coverage completed ratio, calibrating ratio, energy efficiency, and task fairness.

Index Terms—Mobile crowd sensing, UAV, multi-task allocation, reinforcement learning.

1 INTRODUCTION

The rapid development of ubiquitous mobile devices equipped with plenty of sensors over the past few years has given rise to mobile crowd sensing (MCS) [1]–[3]. Compared with traditional sensor networks, MCS reduces the deployment and maintenance cost by outsourcing sensing tasks to individual participants [4]–[6]. In recent years, various MCS applications have been developed in different fields, such as indoor localization [7], public event reporting [8], road and traffic monitoring [9]. Although the MCS campaign provides useful information for several purposes, it still faces crucial challenges to overcome, one of which is how to allocate tasks to meet their requirements, e.g., coverage and sensing data quality requirements.

Sensing coverage is of paramount importance in both the spatial and temporal dimensions [10]. For example, road traffic monitoring typically requires continuous sensing of

H. Gao and J. Feng contributed equally to this work.

interested road segments [11]. As another example, air quality sensing applications usually require collecting the air quality measurements from specific geographic areas in different time instances for analytical and predictive purposes [12]. In real practice, human participants' self-planned trajectories may fail to satisfy the platform's spatio-temporal coverage demand [13].

Collecting high quality sensing data is a fundamental requirement to guarantee the success of MCS applications [14]. The data quality measures the degree of deviation to the ground truth and is sometimes defined as data noise [15]. Many factors can influence the sensing data quality, one of which is the strategy of human participants recruitment [16], [17]. It is a common challenge for most MCS applications to identify credible human participants, then to motive them to collect and contribute high quality data [18]. Unfortunately, as human participants may report less valuable sensing data when trying to minimize their efforts, or have no idea how to collect valuable data, it is difficult for the MCS applications to guarantee valuable service [19]. Another factor that influences the sensing data quality is the measurement error, which depends on whether the sensors have been calibrated [20]. Although the equipped sensors generally did device-level calibration before deployment, such a calibration scheme fails to account for the post-deployment factors, e.g., non-ideal environmental conditions and hardware aging [21]. In a word, sensors may suffer from noise and drift over time. And data sensed from the noise or drift suffered sensors may not meet the accuracy requirement [22]. Thus, these sensors require re-calibration to maintain data accuracy, which is a

This work was supported in part by the National Natural Science Foundation of China under Grant 62002025 and 61802022, in part by the Key research and development projects of Hebei Province under Grant 21310102D, and in part by the Academy of Finland under Grant 317432 and 318937.

H. Gao is with School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China. (E-mail: gaohui786@bupt.edu.cn)

Y. Xiao (Corresponding author) is with Department of Communications and Networking, Aalto University, 02150 Espoo, Finland (E-mail: yu.xiao@aalto.fi).

J. Feng, B. Zhang and W. Wang (Corresponding author) are with the State Key Laboratory of Networking and Switching Technology, School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China. (E-mail: {fengjianhao, zbo, wdwang}@bupt.edu.cn)

fundamental problem in wireless sensor networks.

Previous works mostly considered the single-task allocation scenario [23], [24], where an available human participant is associated with one task at a time. In this paradigm, if a human participant is willing to undertake multiple tasks for rewards, he/she has to wait and interact with the MCS platform for multiple rounds of assignments. Furthermore, as more and more sensing tasks may be time sensitive (such as traffic dynamic monitoring and pollution monitoring at specified locations and time intervals), it is almost indispensable to develop a generic mechanism supporting multiple concurrent MCS task assignments, in order to meet the requirements of all tasks [25], [26].

Recently, unmanned aerial vehicles (UAVs) have been considered to bring a new dimension into MCS [27]–[29]. Particularly, UAVs allow autonomous MCS due to the capability of fast deployment and controllable mobility. Furthermore, as the UAVs could be maintained frequently by the MCS staff, it is easier to calibrate UAVs' sensors than that of human participants, data contributed by them are more accurate and credible. It is a better option to only employ UAVs to perform sensing tasks. However, as the UAVs are forbidden in some cities, e.g., Beijing, Seoul and Washington D.C., human participants are still needed to keep MCS campaigns running.

Here we focus on the overall scenario illustrated in Fig. 1. Imagine that several MCS tasks need to utilize sensing information in a region, e.g., air quality [30], noise level [31], traffic status [32]. For each task, there are several points of interest (PoIs) spread over the region that needs to be sensed by human participants or UAVs. The platform could offer different amounts of rewards for each PoI of a task. A human participant (hereafter referred to simply as "participant") who is interested in the task could apply for the task. At the same time, the UAVs who act as the supplementary part begin to cruise in order to contribute sensing data or calibrate participants' sensing data when they meet participants.

Despite the attractive advantages, UAV-assisted MCS systems face a major challenge: the on-board battery capacity of UAVs imposes a limitation on their endurance capability and performance. Hence, energy efficiency is a critical requirement for such UAV-assisted MCS systems [33]. In another word, the trajectory of each UAV should be scheduled carefully considering the locations of participants, other UAVs and PoIs, when it senses or calibrates, to avoid energy waste. Then how to allocate tasks to participants and UAVs simultaneously has become the primary issue. As the number of participants is much more than that of UAVs, an offered reward method should be well designed in order to recruit enough participants. On the other hand, the UAVs complement the participants. The trajectory of each UAV should be planned with the requirement of avoiding resource waste that a place is sensed by both participants and UAVs, but the sensing coverage should be considered.

In this paper, we propose a UAV-assisted multi-task allocation method for MCS systems with the purpose of meeting task coverage and sensing data quality requirements. The method consists of two parts, one of which is an online incentive mechanism designed for recruiting credible participants. The mechanism first calculates the maximum



Fig. 1. The overall scenario of the UAV-assisted MCS campaign.

offered reward according to the sensing data collection condition and remaining budget. Then a task recommendation method is proposed to recruit credible participants. The other part is designed for scheduling UAV trajectories to contribute data from rarely sensed PoIs. The scheduled trajectories should avoid to sense PoIs that have been or are going to be sensed by participants or other UAVs in order to save energy. We leverage a deep reinforcement learning method to schedule trajectories for UAVs. Furthermore, UAVs are also used to calibrate sensing data contributed by participants. Here the system-level calibration is employed which aims to optimize the overall system performance by tuning the sensing parameters of all the sensors in an MCS network.

The main contribution of this paper is summarized:

- We develop UMA, a multi-task allocation scheme that jointly optimizes the sensing coverage and data quality. The UMA allocates tasks to human participants and UAVs together, with the purpose of collecting high quality sensing data under task deadlines and budget constraints.
- In order to allocate tasks to human participants, we design a learning-based online incentive mechanism that consists of a reward allocation step and a task recommendation step. The mechanism learns the reward offering strategy based on the law of supply and demand, in order to maximize the number of participants while guaranteeing the coverage requirement and budget constraint.
- We propose a deep reinforcement learning based trajectory scheduling mechanism for UAVs, with the purpose of meeting coverage requirements of all tasks. The mechanism first takes locations of participants, other UAVs and PoIs into account, then schedules routes for each UAV to perform tasks efficiently.
- We perform extensive simulations on four real data sets. Compared with four task allocation methods, the effectiveness, robustness and superiority of the proposed algorithm have been extensively evaluated in terms of diverse metrics.

The rest of this paper is organized as follows. We discuss related research efforts in Section 2. The system model is described in Section 3. We introduce the online incentive mechanism in Section 4. And the UAV trajectory scheduling and data calibration mechanism is described in Section 5. We present the simulation results in Section 6. Finally, we conclude the paper in Section 7.

2 RELATED WORK

In this section, we review the related literature covering four topics: task allocation methods for MCS, incentive mechanism for MCS, learning-assisted MCS and systemlevel calibration methods.

2.1 Task allocation methods for MCS

State-of-the-art research in task allocation methods for MCS can be divided into two categories, i.e., single task allocation and multi-task allocation methods. Zhu et al. [34] proposed a single task allocation method that reduced the total costs and improved the sensing data quality. The method consisted of three steps that modeled information, estimated cost and allocated task. Zhao et al. [35] argued that the platform did not know participants' ability to perform tasks in advance, thus they proposed a single task allocation method that modeled participant recruitment as a multi-armed bandit game. Authors in [36] considered multi-dimensional task diversity to design a task allocation method. They formulated the platform-centric and participant-centric auction incentive mechanisms to recruit participants and compute payments. Wu et al. [37] designed a task recommend system that recommended tasks to participants based on their preference and reliability. Authors in [38] proposed a task allocation method that recommended a task to participants based on their preferences and reliability levels. Wang et al. [39] argued that tasks of MCS were usually time-sensitive and location-dependent. Therefore, they proposed a task allocation task method that took task information, such as time and location, into consideration. Authors in [40] investigated a task allocation problem by considering the competition of participants for tasks. They employed the congestion game theory to improve participant satisfaction by considering participant benefit, preference and designed a competition congestion metric. Wang et al. [41] leveraged the social network to recruit participants then allocate them tasks. The platform first selected several participants then influenced other participants using the influence propagation of the social network.

Yucel *et al.* [42] proposed a multi-task allocation method that took participant preferences into account, which employed Matching Theory to find the matching between participants and tasks. Wang *et al.* [15] argued that though the overall utility of multiple tasks is optimized, the sensing quality of individual task might poor. To deal with this problem, the authors proposed a multi-task allocation method that introduced a quality threshold for every single task. Authors in [43] proposed an online multi-task allocation method that updated the task available list for each participant in real-time. Dai *et al.* [44] designed a many-to-many matching algorithm to deal with the multi-task allocation problem, which took participants' requested rewards and sensing data quality into account.

2.2 Incentive mechanisms for MCS

As we know, participants need a reward to incentivize them to contribute sensing data. For location-constrained crowd sensing, Restuccia et al. [45] proposed that the capability of participants to execute sensing tasks depended on their mobility pattern, which was often uncertain. They designed an incentive mechanism that employed reverse auction to recruit participants with uncertain mobility. Xu et al. [46] presented a vehicular location-constrained crowd sensing system. The system incentivized the participants to match the sensing distribution of the sampled data to the desired target distribution with a limited budget. They formulated the incentivizing problem as a knapsack problem and proposed an algorithm named iLOCuS to solve the problem. Fan et al. [47] proposed a joint trajectory scheduling and incentive mechanism for spatio-temporal crowd sensing systems. They designed an online incentive mechanism that decided whether to recruit a participant when he/she asked to contribute sensing data. Hu et al. [48] designed a market-based incentive mechanism, which paid participants monthly or immediately through blockchain. Authors employed the Stackelberg game approach to analyze participants and task publishers' incentive strategies. The authors in [49] proposed an incentive mechanism that formulated a Stackelberg game method to model the interactions among the platforms and participants. Zhang et al. [50] formulated the incentive model with maximizing the reliability of collected sensing data and task coverage. Authors in [51] investigated the problem of online incentive mechanisms by considering time-sensitive tasks. They proposed a method to determine a time-dependent threshold to select participants and calculate payments.

2.3 Learning-assisted MCS

Machine learning techniques have been a new trend to optimize the MCS campaigns. For example, Zhu *et al.* [52] proposed an online participant selection method. The method first employed a deep learning method to predict participant mobility, then a greedy online algorithm was proposed to recruit participants. Authors in [53] proposed a deep reinforcement learning method to assign sensing tasks to participants with the purpose of collecting high quality sensing data and saving sensing costs. Hu *et al.* [54] proposed a task allocation method that employed a reinforcement learning approach to jointly consider both the previous coverage and participant current mobility predictability.

In this paper, we employ the deep reinforcement learning method to schedule the UAVs' trajectories, which has recently attracted much attention from both industry and academia. In a pioneering work [55], the deep Q-learning (DQL) method, a reinforcement learning framework, was proposed to improve learning stability. Lillicrap *et al.* [56] presented an actor-critic, model-free algorithm based on the deterministic policy gradient that could operate over a continuous action space. Based on this, Lowe *et al.* [57] presented another actor-critic method. The method considered action policies of other agents and was able to successfully learn policies that required complex multi-agent coordination.

2.4 System-level calibration methods

Several approaches to sensor calibration have been presented in the literature. For example, authors in [58] in-

TABLE 1 List of Important Notations

Notation	Explantation
$\mathcal{T}, B^t, \mathcal{P}^t, \mathcal{K}^t$	Set of tasks, budget of one task, set of PoIs, set of timeslots
η_p^t , $\hat{\eta}_p^t(k^t)$	No. of requested pieces of sensing data, No. of collected pieces of sensing data until k^t
$\mathscr{L}_i, \mathscr{L}_i^h \ c_i^k$, $c(\mathscr{L}_i)$	Set of sensed PoIs, set of contributed high quality sensing data, the required reward during time slot k and the total offered reward
$v_j(k^t)$, $a_j(k^t)$	Vector velocity and acceleration of UAV j
$e_j(k^t)$, E_j	Energy consumption and battery capacity of UAV <i>j</i>
\mathscr{L}_j , \mathscr{L}_j^m	Set of PoIs collected by UAV j , set of calibration
c^t , $c_p(k^t)$, $c_p^f(k^t)$	Basic reward, floating reward and the maximum offered reward in PoI p at timeslat kt
$P_i^t(k^t)$	Probability of high quality sensing data contributed by participant i

vestigated how the fusion of data taken by sensor arrays could improve the calibration process. Lin *et al.* [59] proposed a two-phase data calibration method and employed two methods to train these two parts, respectively. The work required a large amount of training data to learn the calibration curve and thus could not provide real time data results, which could be summarized as the offline calibration method. Differing from the offline calibration methods, we propose an online approach, which leverages the historical calibration curve of mobile sensors to reduce calibrating times and improve data accuracy.

Compared with the existing research mentioned above, we propose a task allocation method for MCS systems that employs the UAVs and participants to jointly collect data. The coexistence of multiple concurrent tasks is taken into consideration, which makes our system more complex but more efficient. Fig. 2 illustrates the framework of our proposed approaches. In order to meet the coverage and data quality requirements of tasks, we design an incentive mechanism that calculates the maximum offered reward of each PoI and allocates the task to participants. If a PoI is rarely sensed, a higher price will be given to encourage participants to contribute sensing data. A task recommendation method helps the platform recommend tasks to credible participants who may contribute high quality sensing data. On the other hand, differing from all the above research work, the UAVs are not only employed as the supplementary part to contribute data from PoIs which are rarely sensed by participants or other UAVs, but also they are used to calibrate low precision sensor collection data of participants. The trajectory of each UAV is scheduled efficiently considering the locations of participants, other UAVs and the rarely sensed PoIs, with the purpose of avoiding energy waste. In addition, we propose a calibration approach, which could reduce the meeting times to calibrate.

3 SYSTEM MODEL

We consider an MCS system that provides services for smart cities every day. There are a set of T concurrent tasks

4



Fig. 2. The workflow of proposed approaches.

denoted by $\mathcal{T} = \{t|t_1, t_2, \ldots, t_T\}$. Each task is associated with certain task budget B^t and a set of P PoIs that is denoted by $\mathcal{P}^t = \{p^t | p_1^t, p_2^t, \ldots, p_P^t\}$. Furthermore, the whole sensing campaign is divided into K time-slots with equal duration Λ , that $\mathcal{K}^t = \{k^t | k_1^t, k_2^t, \ldots, k_K^t\}$. Normally, the platform selects several pieces of sensing data contributed by participants for one PoI to get more accurate results. Each PoI needs to be sensed no more than η_p^t times by participants or 1 time by a UAV during one time-slot, and $\hat{\eta}_p^t(k^t)$ denotes the number of times PoI p has been sensed until time-slot k^t .

For a participant *i* who prepares to perform the task in one time-slot, he/she will claim his/her location and requested a reward at the very beginning of the time-slot. The platform will select several participants according to their requested rewards and locations, and the task requirement. Each participant has a requested reward which is denoted by c_i^t for a task $t \in \mathcal{T}$. The final set of PoIs sensed by participant *i* is denoted by $\mathscr{L}_i = \left\{ x_i(k^t, p^t), k^t \in \mathscr{K}^t, p^t \in \mathscr{P}^t \right\},\$ where $x_i(k^t, p^t) = 1$ denotes participant *i* contributes a piece of sensing data for task t at PoI p^t when timeslot is k^t , otherwise $x_i(k^t, p^t) = 0$. And the corresponding final reward of the recruited participant i is denoted by $c(\mathcal{L}_i)$. The set of high quality sensing data is denoted by $\mathscr{L}_{i}^{h} = \left\{ y_{i}(k^{t}, p^{t}), k^{t} \in \mathscr{K}^{t}, p^{t} \in \mathscr{P}^{t} \right\}$, where $y_{i}(k^{t}, p^{t}) = 1$ denotes participant *i* contributes a piece of high quality sensing data for task t at PoI p^t when time-slot is k^t , otherwise $y_i(k^t, p^t) = 0$.

We consider sensing tasks as that all UAVs fly around to cover PoIs. In the beginning, a UAV j that is with a fully charged battery moves with the vector velocity $\begin{array}{l} v_j(k^t) = \left(\theta_j^v(k^t), d_j^v(k^t)\right)_{j \in \mathscr{J}'}, \text{ where } |v_j(k^t)| \in [0, |v_{max}|], \\ \theta_j(k^t) \in [0, 2\pi) \text{ is a direction and } d_j(k^t) \text{ is speed. The vector velocity is controlled by the vector acceleration, which is denoted by <math>a_j(k^t) = (\theta_j^a(k^t), d_j^a(k^t))_{j \in \mathscr{J}}. \text{ Here, we consider the energy consumption } e_j(k^t), \text{ which is simply proportional to flying distance, i.e., } e_j(k^t) = \gamma d_j(k^t). \text{ And the battery capacity is denoted by } \mathscr{L}_j. \text{ The final coverage set is denoted by } \mathscr{L}_j = \left\{ x_j(k^t, p^t), k^t \in \mathscr{K}^t, p^t \in \mathscr{P}^t \right\}, \text{ where } x_j(k^t, p^t) = 1 \text{ denotes a UAV } j \text{ contributes a piece of high quality sensing data for task } t \text{ at PoI } p^t \text{ when timeslot is } k^t, \text{ otherwise } x_j(k^t, p^t) = 0. \text{ The set of calibration is denoted by } \mathscr{L}_j^m = \left\{ y_j(k^t, i), k^t \in \mathscr{K}^t \right\}, \text{ where } y_j(k^t, i) = 1 \text{ denotes sensing data for participant } i \text{ when timeslot is } k^t, \text{ otherwise } y_j(k^t, i) = 0. \end{array}$

The frequently used notations are summarized in Table 1.

The target of this paper is to maximize the number of collected high quality sensing data in order to meet the coverage requirement under the consideration of the budget and UAV energy constraints, which can be formulated as

$$\begin{array}{l} \text{maximize:} \quad \sum_{t=1}^{T} \frac{\left| \bigcup_{i \in \{1,2,\dots,I\}} \mathscr{L}_{i}^{h} \right| + \left| \bigcup_{j \in \{1,2,\dots,J\}} \mathscr{L}_{j} \right| * \eta_{p}^{t} \\ \eta_{p}^{t} * |\mathscr{P}^{t}| * |\mathscr{K}^{t}| \\ \text{subject to:} \quad \sum_{t=1}^{T} \sum_{k^{t}=1}^{K} \sum_{i=1}^{I} c(\mathscr{L}_{i}) \leq \sum_{t=1}^{T} B^{t}, \\ \sum_{t=1}^{T} \sum_{k^{t}=1}^{K} e_{j}(k^{t}) \leq E_{j}, \end{array}$$

$$(1)$$

where I and J is the number of recruiting participants and

UAVs, respectively. $\left| \bigcup_{i \in \{1,2,\dots,I\}} \mathscr{L}_{i}^{h} \right| + \left| \bigcup_{j \in \{1,2,\dots,J\}} \mathscr{L}_{j} \right| * \eta_{p}^{t}$ denotes the total contributed high quality sensing data for task $t. \eta_{p}^{t}$ is the number of required times sensed from a PoI. $\eta_{p}^{t} * |\mathscr{P}^{t}| * |\mathscr{K}^{t}|$ denotes the number of required high quality sensing data of task t. The range of the problem formulation is [0, 1], where 0 means there is no high quality sensing data contributed. On the contrary, 1 means the coverage requirement of the task has been fully satisfied.

Lemma 1. The target of this paper is an NP-hard problem.

Proof. The budget limited maximum coverage problem has been proved to be an NP-complete problem, which can be described as follows: given a collection of sets $\mathscr{S} = \{\mathscr{S}_1, \mathscr{S}_2, \ldots, \mathscr{S}_m\}$ with associated costs $\{c_i\}_{i=1}^m$ is defined over a domain of weighted elements $\mathscr{X} = \{x_1, x_2, \ldots, x_n\}$ with associated weights $\{\omega_i\}_{i=1}^n$. The goal is to find a collect of sets $\mathscr{S}' \subseteq \mathscr{S}$, such that the total cost of sets in \mathscr{S}' does not exceed L, and the total weight of elements covered by \mathscr{S}' is maximized.

We prove the lemma by reducing the previous problem to an instance of (1). Imagine there is only one task performed one time-slot. And the task requires one piece of data from a PoI, that T = 1, K = 1 and $\eta_p = 1$. For

maximize:
$$\frac{\left|\bigcup_{i \in \{1,2,\dots,I\}} \mathscr{L}_{i}^{h}\right|}{|\mathscr{P}|}$$
(2)
subject to:
$$\sum_{i=1}^{I} c(\mathscr{L}_{i}) \leq B.$$

For a collection PoI sets of $\mathscr{L} = \{\mathscr{L}^1, \mathscr{L}^2, \dots, \mathscr{L}^m\}$. Each element $\mathscr{L}^m \in \mathscr{L}$ contains several numbers of PoIs, and the domain of PoIs is denoted by $\mathscr{P} = \{p_1, p_2, \dots, p_n\}$ with associated maximum offered reward $\{c_p^f\}_{p=1}^n$. The total offered reward of \mathscr{L}^m is denoted by c^m . The target of (2) is to find a set of participants under budget constraints with the purpose of covering the maximum number of PoIs. In another word, the target is to find a collection of PoI sets $\mathscr{L}' \subseteq \mathscr{L}$, where $\mathscr{L}' = \bigcup_{i \in \{1, 2, \dots, I\}} \mathscr{L}_i^h$, in order to maximize $|\mathscr{L}'|/|\mathscr{P}|$ under the budget B.

As demonstrated above, the budget limited maximum coverage problem is as complex as the simplified (2), which means that the proposed maximum target (1) is also NP-complete, and then completes the proof. \Box

4 ONLINE INCENTIVE MECHANISM

we design an online incentive mechanism that consists of a reward allocation step and a task recommendation step. For the reward allocation step, we decide the maximum offered reward considering the data collecting situation. For the task recommendation step, we predict which tasks are good for a participant for performing.

4.1 Maximum offered reward decision mechanism

Following the former work [60], the reward paid for sensing acts as a signal to reflect sensing data supply and demand, which depends on the demand of the platform and the supply of participants. Here we employ the maximum offered reward to decide the maximum reward offered to the participants. The maximum offered reward of every time-slot is denoted by $c_p^f(k)$, which consists of the basic offered reward c^t and floating reward $c_p(k)$. For the sake of simplicity, hereafter we drop the task index t and treat all tasks equally.

The basic offered reward indicates the ideal fixed cost if the amount of requested sensing data is collected under budget constraints. The floating reward exists to make the maximum offered reward higher or lower, based on the number of participants or data collected consideration. For example, the maximum offered reward could be higher for a PoI, if there is less amount of sensing data collected.

The deep reinforcement learning based method is employed to calculate the floating reward offered for one piece of sensing data in every PoI at each timeslot. Coarsely speaking, the proposed method involves a decision agent that repeatedly observes the current states of the participant recruitment, then takes an action among the available actions allowed in that state. After, the agent will transfer to a new state and obtain the corresponding reward. 1) State Space: $\mathcal{N} \triangleq \{n^k = (N_1^k, N_2^k)\}$ denotes the state that indicates whether the participant is recruited or not.

2) Action Space: $\mathscr{A} \triangleq \{a^k | a \in \mathscr{V}\}$ denotes the action set.

3) Probability Distribution and State Transition: $F : \mathcal{N} \times \mathcal{A} \times \mathcal{N} \rightarrow [0,1]$ denotes the probability distribution $P\{n^{k+1}|n^k, \{a^k\}_{k\in\mathscr{X}}\}$ of a state transition, in which the current state is n^k and when action a^k is chosen, the state is transitioned to a new state n^{k+1} .

4) Reward Function: $\mathcal{N} \times \mathcal{A} \to \mathbb{R}$ expresses the expected immediate reward received after the state is transitioned from n^k to n^{k+1} , due to taking the action a^k , which is defined as: $r^k = e^{a^k} / \sum_{Z=1}^{Z} e^{v_z}$. Here we employ the softmax value to calculate the reward.

5) Problem Formulation: When state transition F and reward function r_p^k , $p \in \mathcal{P}$, is predetermined, for each timeslot k, our problem can be formulated as

$$\begin{aligned} Q_p(\boldsymbol{n}^k) &= \max_{a_p^k} \left[r_p^k(\boldsymbol{n}^k, a_p^k) \right. \\ &+ \gamma \int_{\boldsymbol{n}^k \in \mathcal{N}} F(\boldsymbol{n}^k, a_p^k, \boldsymbol{n}^{k+1}) Q_p(\boldsymbol{n}^{k+1}) \right], \end{aligned}$$

and the optimal strategies of the floating reward is given by

$$c_{p}(k) = \arg \max_{a_{p}^{k}} \left[r_{p}^{k}(n^{k}, a_{p}^{k}) + \gamma \int_{n^{s} \in \mathcal{N}} F(n^{k}, a_{p}^{k}, n^{k+1}) Q_{p}(n^{k+1}) \right].$$
(3)

Based on (3), the floating reward can be decided in PoI p at time-slot k. Here a special phenomenon is also considered, that the budget could not afford the sum of maximum offered reward of all PoIs. For this reason, some rarely sensed PoIs are more important. Therefore, a method is needed to help the platform recruit participants preferentially from these PoIs under the limited budget. Here the Shapley method is employed to identify which PoIs are important, which is shown as

$$\lambda_p(k) = \sum_{\mathscr{P}' \subseteq \mathscr{P} \setminus \{p\}} \frac{|\mathscr{P}'|! (|\mathscr{P}| - |\mathscr{P}'| - 1)!}{|\mathscr{P}|!} f(\mathscr{P}' \bigcup \{p\}), \quad (4)$$

where $f(\mathscr{P}' \bigcup \{p\})$ is the marginal value, which is shown as

$$\begin{split} f(\mathscr{P}' \bigcup\{p\}) &= \\ \sum_{\tau=1}^{\eta_{\mathcal{P}}(k)} \left(\left(1 - \frac{\| [\eta_1(k), \eta_2(k), \dots, \eta_{|\mathscr{P}'|}(k), \eta_p(k) - \tau] \|_F}{\eta \sqrt{|\mathscr{P}' \bigcup\{p\}|}} \right) \\ &- \left(1 - \frac{\| [\eta_1(k), \eta_2(k), \dots, \eta_{|\mathscr{P}'|}(k)] \|_F}{\eta \sqrt{|\mathscr{P}'|}} \right) \right), \end{split}$$

where $\eta_p(k) = \eta_p - \hat{\eta}_p(k), p \in \mathscr{P}$ is the number of pieces of sensing data that has not been collected yet, and $\|\cdot\|_F$ is the Frobenius norm, which is mathematically used to measure the spatial length of a matrix, to quantify the difference between the required and attained values.

The maximum offered reward decision mechanism is presented in Algorithm 1. Firstly the basic reward is calculated in Line 2. Then the mechanism calculates the floating reward for each PoI at one timeslot, and the maximum offered reward (Line 4 - Line 8). Finally, the Shapley method is used to rank all of PoIs in descending order (Line 9 - Line 10).

Algorithm 1 Maximum Offered Reward Decision Mechanism

Input: Budget B^t , sensing requirement η_p , time-slot \mathscr{K}^t **Output:** The new sensing ranges of PoIs and the maximum offered reward of every PoI $c_p^f(k^t)$ at timeslot k^t

1: /*Calculating the basic offered reward*/

2: $c^t = B^t / (P * \eta_p);$

3: /*iterating through all of PoIs*/

4: for l = 1, ..., P do
5: Calculate the floating reward candidate by (3);

6: /*Calculating the maximum offered reward*/

7: $c_p^f(k^t) = c^t + c_p(k^t);$

8: end for

9: Calculate Shapley value $\lambda_p(k^t)$ by (4); 10: Rank PoIs using $\lambda_p(k^t)$ in descending order;

4.2 Task recommendation method

Several tasks need to be issued along a participant's trajectory. Before recommending the participant tasks, it is essential to predict which tasks are good for him/her, with the purpose of collecting high quality sensing data. The quality of data is denoted by $q_{i,n}^t$, which is contributed by a participant *i* for task $t \in \mathcal{T}$ in the *n*th time. Here, we also drop the task index *t* for the same reason. Following our previous work [60], the quality of sensing data contributed by a participant is modeled as a semi-Markov with discrete time.

The kernel part of semi-Markov is defined in (5). $W_i^{uh}(k)$ denotes the probability that a participant *i* contributes high quality sensing data in *n*th time at a timeslot *k*, given he/she contributed unusable quality sensing data in the (n-1)th time. $f_i(k) \leq \Lambda$ means the participant contributes sensing data in the time duration Λ . Here we assume that a participant will contribute sensing data before the end of the timeslot.

$$W_i^{uh}(k) = P(q_{i,n} = h, f_i(k) \le \Lambda | q_{i,n-1} = u).$$
 (5)

Next, the probability that a participant *i* contributes high quality sensing data at the *n*th time while he/she contributes unusable quality data at last time, before time duration Λ , is denoted by $Z_i^{uh}(\cdot)$, which is shown as

$$Z_{i}^{uh}(\Lambda) = P(f_{i}(k) \leq \Lambda | q_{i,n} = h, q_{i,n-1} = u)$$

= $\sum_{x=1}^{\Lambda} P(f_{i}(k) = x | q_{i,n} = h, q_{i,n-1} = u).$ (6)

The probability that a participant *i* contributes high quality sensing data at the *n*th time, when he/she contributes unusable quality data at the (n - 1)th time is calculated by

$$P_i^{uh} = P(q_{i,n} = h | q_{i,n-1} = u) = \frac{num_i^{uh}}{num_i^u}, \tag{7}$$

where num_i^{uh} is the number of times data quality contributed from unusable to high quality, while num_i^u is the number of times unusable data contributed.

We rewrite (5) based on (6) and (7), which is shown as

$$W_{i}^{un}(k) = P(q_{i,n} = h, f_{i}(k) \le \Lambda | q_{i,n-1} = u)$$

= $Z_{i}^{uh}(\Lambda) P_{i}^{uh}.$ (8)

Algorithm 2 Task Recommendation Mechanism
Input: Budget B^t , uncollected data $\eta_p(k^t)$, time-slot \mathscr{K}^t Output: Recommended PoI set \mathscr{L}'_i
1: $k^t = 1;$
2: /*iterating through all of time-slots*/
3: while $k^t \leq K \operatorname{do}$
4: Calculating the new sensing range of PoIs with the maximum offered
rewards in Algorithm 1;
5: $p^t = 1;$
6: /*Starting to allocate tasks*/
7: while $p^t \leq P$ && p^t is in the sensing range of participant <i>i</i> do
8: if $B^t - c_p^f(k^t) \ge 0 \& \& \eta_p(k^t) > 0$ then
9: Calculate $P_i^t(k^t)$ by (9);
10: $num = 1$ with probability $P_i^t(k^t)$;
11: if $num == 1$ then
12: $\{k^t, p^t\} \to \mathcal{Z}'_i;$
13: end if
14: end if
15: /*For the next PoI*/
16: $p^t = p^t + 1;$
17: end while
18: /*For the next time-slot*/
19: $k^t = k^t + 1;$
20: end while

Based on (8), the probability that a participant i contributes high quality sensing data at a timeslot k is shown as

$$P_i(k) = \frac{W_i^{uh}(k) + W_i^{hh}(k)}{W_i^{uh}(k) + W_i^{uu}(k) + W_i^{hu}(k) + W_i^{hh}(k)}.$$
 (9)

The task recommendation method is shown in Algorithm 2, correspondingly we describe the main processes as follows.

Step 1: At the beginning of each time-slot k^t , the maximum offered reward of every PoI is calculated by Algorithm 1 in Line 4, along with the new sensing range.

Step 2: The platform begins to allocate tasks to participants. For a PoI of each task which is in the sensing range of participant *i*. If the rest budget could afford the maximum offered reward, and there is still several amount of sensing data needed to be collected, then the PoI has a chance to be sensed (see Line 8). After that, the platform calculates probability $P_i^t(k^t)$ to decide whether the PoI p^t could be performed by the participant *i*. If the result is positive, the platform collects the PoI for recommendation (see Line 9-13).

Step 3: The crowd sensing campaign will be ended when one of the three conditions is met:

- The last sensing time-slot is finished.
- The required amount of sensing data is met.
- The budget is exhausted.

5 UAV TRAJECTORY SCHEDULING AND DATA CALIBRATION MECHANISMS

In this section, we first introduce a UAV trajectory scheduling mechanism, which directs the UAVs to contribute sensing data from PoIs which are rarely accessed by participants. Then a sensing data calibration method is proposed to improve the quality of data collected by participants. It is worth noting that both trajectory scheduling and data calibration methods are calculated by the platform which is typically run on a cloud server. The UAVs receive and follow commands. More details are introduced in the sections below.

5.1 Learning-based UAV trajectory scheduling mechanism

As we mentioned in Section 1, there are two purposes for UAVs, i.e., data collection and calibration. Here we present the proposed Learning-based UAV trajectory scheduling mechanism for achieving these two purposes. We formulate our problem as a Markov Decision Process. It is noted that we also drop the task index t for the same reason.

1) State space and observation space: $S = \{s^k | k \in \mathcal{R}\}$ denotes the state set of an MDP, where s^k consists of four parts. The first part is the state set of all UAVs that $\{(x_j^k, y_j^k), e_j^k\}_{j \in \mathcal{I}}$, where (x_j^k, y_j^k) denotes the current position of a UAV j in time k. $e_j^k \in [0\%, 100\%]$ denotes the remaining energy of a UAV j that is expressed by a percentage. The second part is the state set of all participants that $\{(x_i^k, y_i^k), m_i^k\}_{i \in \mathcal{I}}$, where (x_i^k, y_i^k) is the position of a participant i in time k, and m_i^k is the accumulated number of calibrating times with UAVs. The third part is the obstacle position that $\{(x_o, y_o)\}$ that UAVs should avoid hitting. The forth part is the state set of all PoIs that $\{(x_p, y_p), f_p^k\}$, where (x_p, y_p) denotes the position of PoI p. And f_p^k denotes the sensing requirement completion percentage of PoI p in time k that can be expressed as

$$f_p^k = \begin{cases} \frac{\hat{\eta}_p(k)}{\eta_p}, & \text{if } \hat{\eta}_p(k) \le \eta_p \\ 1, & \text{otherwise} \end{cases}$$

However, each UAV only knows a part of information of the state space which is called observation. The observation space is denoted by $\mathcal{O}^k = \{o_j^k | j \in \mathcal{J}, o_j^k \subseteq s^k\}$.

2) Action space: The action set is denoted by $\mathscr{A} = \{a_j^k = (\theta_j^a(k), d_j^a(k))_{j \in \mathscr{F}} | \theta_j^a(k) \in [0, 2\pi), d_j^a(k) \in [0, d_{max}] \}$, where $\theta_j^a(k)$ and $d_j^a(k)$ is direction and acceleration value of UAV j at time k, respectively.

3) Probability distribution and state transition: $F : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ denotes the probability distribution $P\{s^{k+1}|s^k, \{a_j^k\}_{j \in \mathcal{F}}\}$ of a state transition, in which the current state is s^k and when action a_j^k is chosen, the state is transitioned to a new state s^{k+1} .

4) Reward function: $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$ expresses the expected immediate reward received after the state is transitioned from s^k to s^{k+1} , due to taking the action $\{a_j^k\}_{j \in \mathcal{J}}$, which is defined as: $r_j^k = \left(\mu o_j^k + (1-\mu)l_j^k\right)/e_j(k) - g_j^k$, where o_j^k is the amount of data collected by a UAV j at timeslot k, l_j^k is the number of calibrating (meeting) times with participants. And $\mu \in [0,1]$ is a parameter to decide the higher priority between data collection and calibration campaign. The platform can adjust the value of μ for adapting to different scenarios and task requirements. When UAV hits an obstacle or moves out of the area border in the timeslot k, the penalty g_j^k should be taken off. Therefore, the reward definition $r_j^k, \forall j$ can be considered to incorporate three objectives, data collection amount, times of calibration and energy consumption.

5) Problem formulation: When state transition F and reward function r_i^k , $k \in \mathcal{K}$, $j \in \mathcal{J}$ is predetermined, for each stage s, our problem can be formulated as

$$\begin{split} Q_{j}(s^{k}) &= \max_{a_{j}^{k}} \left[r_{j}^{k}(s^{k}, a_{j}^{k}) \right. \\ &+ \gamma \int_{s^{k+1} \in \mathcal{S}} F(s^{k}, a_{j}^{k}, s^{k+1}) Q_{j}(s^{k+1}) \right], \end{split}$$

and the optimal strategies of a UAV j is given by

$$\pi_j = \arg \max_{a_j^k} \left[r_j^k(s^k, a_j^k) + \gamma \int_{s^{k+1} \in \mathcal{S}} F(s^k, a_j^k, s^{k+1}) Q_j(s^{k+1}) \right],$$

where $\gamma \in (0, 1)$ represents the discount factor, which shows the importance between the future reward and present reward.

Obviously, it is a continued control problem that cannot be solved via the conventional dynamic programming method. In addition, since our scenario is fully distributed as a multi-agent environment, a UAV's reward is affected by the actions of many other UAVs. Hence, we employ the Multi-Agent Deep Deterministic Policy Gradient (MAD-DPG) approach to find the suboptimal solution.

In our proposed solution for the UAV trajectory scheduling problem, each UAV j has its policy decision network, which is divided into three parts: CNN extracts features of the observation o_i^k , actor network decides the action and critic network estimates the action, where the design and implement of actor network and critic network are based on MADDPG. Firstly, we utilize the CNN to extract features from the observation o_i^k , in order to help each UAV to make decisions, including: (a) the related positions with PoIs/obstacles/participants, (b) the distribution of PoIs that do not meet the number of sensing requirements, (c) the distribution of the meeting times with each participant. We next utilize the observation feature results extracted from the CNN, action and reward to train the actor model and critic model. The actor model decides the action of UAV *j* according to the observation, and the critic model will unite with the action of other UAVs and the overall state to estimate its action, aiming to prove the overall reward. It is noted that, in our solution, a UAV has its policy to collect data or calibrate sensors, which is more suitable to our scenario compared with methods that the UAVs use the same policy.

5.2 Sensing data calibration method

As we mentioned in Section 1, sensors suffer from noise and drift over time. Here we adopt the sensing data calibration method to calibrate contributed sensing data from these sensors. It is worth noting that the proposed method is a system-level calibration method that works for data of air quality, environment noise and GPS location, etc. For a task $t \in \mathcal{T}$, inspired by dimension projection [61], we project the measurements of mobile sensors into high dimensional space. Then, we adopt the linear regression model in matrix form as $\boldsymbol{x}_i(\boldsymbol{o}) = \Phi_i(\boldsymbol{o})\boldsymbol{w}_i + \boldsymbol{e}_i(\boldsymbol{o})$, where \boldsymbol{o} is a two dimensional matrix $\boldsymbol{o} \subseteq \mathcal{K} \times \mathcal{P}$. $\Phi_i(\boldsymbol{o}) = [\phi(y_i^p(k)), \{k, p\} \in \boldsymbol{o}]^T$, and $y_i^p(k)$ is the measurement of sensing data contributed by a participant i in PoI p at timeslot k. \boldsymbol{w}_i is the vector of calibration curve of participant *i*, that $\boldsymbol{w}_i = \begin{bmatrix} \omega_1^i \, \omega_2^i \dots \, \omega_P^i \end{bmatrix}^T$. $\boldsymbol{e}_i(\boldsymbol{o})$ is the matrix of noise $\varepsilon_i^p(k)$, that $\boldsymbol{e}_i(\boldsymbol{o}) = \begin{bmatrix} \varepsilon_i^p(k), \{k, p\} \in \boldsymbol{o} \end{bmatrix}^T$.

Let $p(\boldsymbol{w}_i)$ be the prior distribution of \boldsymbol{w}_i , that $p(\boldsymbol{w}_i) = \mathcal{N}(\boldsymbol{w}_i | \mu_i, \Sigma_i)$, where μ_i and Σ_i are the prior mean vector and prior covariance matrix of \boldsymbol{w}_i , respectively. $\mathcal{N}(\cdot)$ denotes the probability density function of Gaussian distribution. The noise $\varepsilon_m^{a,l}(t)$ is also supposed to follow Gaussian distribution, where $p(\varepsilon_i^p(k)) = \mathcal{N}(\varepsilon_i^p(k)|0,\beta_i)$. Then, the probability density function of $\boldsymbol{x}_i(\boldsymbol{o})$ is shown as

$$p(\boldsymbol{x}_{i}(\boldsymbol{o})|\boldsymbol{w}_{i}) = \prod_{\{k,p\}\in\boldsymbol{o}} p(x_{i}^{p}(k)|\boldsymbol{w}_{i})$$
$$= \prod_{\{k,p\}\in\boldsymbol{o}} \mathcal{N}(x_{i}^{p}(k)|\boldsymbol{w}_{i}^{T}\phi(y_{i}^{p}(k)),\beta_{i}),$$

where $\boldsymbol{x}_i(\boldsymbol{o}) = \left[x_i^p(k), \{k, p\} \in \boldsymbol{o}\right]^T$.

Next, we first prove that the posterior mean vector $\hat{\mu}_i$ and covariance matrix $\hat{\Sigma}_i$ of w_i can be updated by Theorem 1 and 2. Then, $\hat{x}_i(o)$ can be estimated by $\hat{x}_i(o) = \Phi_i(o)\hat{\mu}_i$.

Theorem 1. The posterior mean vector $\hat{\mu}_i$ of w_i can be updated by (10).

$$\hat{\mu}_{i} = \left(\frac{1}{\beta_{i}} \Phi_{i}(\boldsymbol{o})^{T} \Phi_{i}(\boldsymbol{o}) + (\Sigma_{i})^{-1}\right)^{-1} \times \left(\frac{1}{\beta_{i}} \Phi_{i}(\boldsymbol{o})^{T} \boldsymbol{x}_{i}(\boldsymbol{o}) + (\Sigma_{i})^{-1} \mu_{i}\right), \\ \forall \boldsymbol{o} \subseteq \mathscr{K} \times \mathscr{P}.$$
(10)

Proof. we denote the posterior distribution of w_i by $p(w_i|x_i(o))$, then we have that $p(w_i|x_i(o)) \propto p(x_i(o)|w_i)p(w_i)$, where $\forall o \subseteq \mathscr{K} \times \mathscr{P}$. The results in the log-likelihood function which is shown as

$$L(\boldsymbol{w}_{i},\boldsymbol{o}) \triangleq \ln p(\boldsymbol{x}_{i}(\boldsymbol{o})|\boldsymbol{w}_{i})p(\boldsymbol{w}_{i})$$

$$= -\frac{|\boldsymbol{o}| + P}{2}\ln 2\pi - \frac{|\boldsymbol{o}|}{2}\beta_{i} - \frac{1}{2}\ln|\boldsymbol{\Sigma}_{i}|$$

$$- \frac{1}{2\beta_{i}}(\boldsymbol{x}_{i}(\boldsymbol{o}) - \Phi_{i}(\boldsymbol{o})\boldsymbol{w}_{i})^{T}(\boldsymbol{x}_{i}(\boldsymbol{o}) - \Phi_{i}(\boldsymbol{o})\boldsymbol{w}_{i})$$

$$- \frac{1}{2}(\boldsymbol{w}_{i} - \mu_{i})^{T}(\boldsymbol{\Sigma}_{i})^{-1}(\boldsymbol{w}_{i} - \mu_{i}),$$

The posterior mean vector of w_i , $\hat{\mu}_i$, can be calculated by $\hat{\mu}_i = \underset{w_i}{\operatorname{arg\,max}} L(w_i, o)$. Taking the derivative of loglikelihood function in respect to w_i that

$$\frac{\partial L(\boldsymbol{w}_{i},\boldsymbol{o})}{\partial \boldsymbol{w}_{i}} = \frac{1}{\beta_{i}} \Phi_{i}(\boldsymbol{o})^{T} \left(\boldsymbol{x}_{i}(\boldsymbol{o}) - \Phi_{i}(\boldsymbol{o})\boldsymbol{w}_{i}\right) - (\Sigma_{i})^{-1} (\boldsymbol{w}_{i} - \mu_{i}),$$

$$\forall \boldsymbol{o} \subseteq \mathscr{K} \times \mathscr{P}.$$
(11)

Theorem 1 is proved, when $\partial L(\boldsymbol{w}_i, \boldsymbol{o}) / \partial \boldsymbol{w}_i = 0.$

Theorem 2. The posterior covariance matrix Σ_i of w_i could be updated by (12).

$$\hat{\Sigma}_{i} = \left(\frac{1}{\beta_{i}} \Phi_{i}(\boldsymbol{o})^{T} \Phi_{i}(\boldsymbol{o}) + (\Sigma_{i})^{-1}\right)^{-1}, \\ \forall \boldsymbol{o} \subseteq \mathscr{K} \times \mathscr{P}.$$
(12)

Proof. According to the Bayesian Cramér-Rao bound [62], the mean square error matrix $\hat{\Sigma}_i$ is bounded from below by

the inverse of the Fisher information matrix $J(w_i)$, which can be formulated as $\hat{\Sigma}_i \succeq J(\boldsymbol{w}_i)^{-1}$, where

$$\hat{\Sigma}_i = E[(\boldsymbol{w}_i - \hat{\mu}_i)(\boldsymbol{w}_i - \hat{\mu}_i)^T],$$

and

$$J(\boldsymbol{w}_i) = E[-\partial_{\boldsymbol{w}_i}^2 \ln p(\boldsymbol{x}_i(\boldsymbol{o}), \boldsymbol{w}_i)],$$

where $\partial_{\boldsymbol{w}_i}^2$ denotes the Laplacian or second-order differential operator with respect to w_i .

Based on the results of (11), we have

$$J(\boldsymbol{w}_{i}) = -E[\partial_{\boldsymbol{w}_{i}}^{2}L(\boldsymbol{w}_{i},\boldsymbol{o})]$$

$$= E[\frac{1}{\beta_{i}}\Phi_{i}(\boldsymbol{o})^{T}\Phi_{i}(\boldsymbol{o}) + (\Sigma_{i})^{-1}]$$

$$= \frac{1}{\beta_{i}}\Phi_{i}(\boldsymbol{o})^{T}\Phi_{i}(\boldsymbol{o}) + (\Sigma_{i})^{-1},$$

$$\forall \boldsymbol{o} \subseteq \mathscr{K} \times \mathscr{P}, \qquad (13)$$

The expectation is in respect to w_i . According to [63], $\hat{\mu}_i$ is the best linear unbiased estimator which can achieve the Cramér-Rao lower bound $J(w_i)$ under linear Gaussian condition. Thus, we have

$$\hat{\Sigma}_i = J(\boldsymbol{w}_i)^{-1} = \left(\frac{1}{\beta_i} \Phi_i(\boldsymbol{o})^T \Phi_i(\boldsymbol{o}) + (\Sigma_i)^{-1}\right)^{-1},$$

h proves Theorem 2.

which proves Theorem 2.

It should be noted that, $\hat{\Sigma}_i$ only shows how well can we estimate w_i . To illustrate how well can we estimate $\boldsymbol{x}_i(\boldsymbol{o})$, we next evaluate the expected value of the mean square error matrix between the ground truth $x_i(o)$ and the estimation $\hat{x}_i(o)$ by Theorem 3.

Theorem 3. The expected value of the mean square error matrix between the ground truth $x_i(o)$ and the estimation of the ground truth $\hat{x}_i(o)$ can be calculated by (14).

$$E\left[\left(\hat{\boldsymbol{x}}_{i}(\boldsymbol{o}) - \boldsymbol{x}_{i}(\boldsymbol{o})\right)^{T}\left(\hat{\boldsymbol{x}}_{i}(\boldsymbol{o}) - \boldsymbol{x}_{i}(\boldsymbol{o})\right)\right]$$

= $\Phi_{i}(\boldsymbol{o})\hat{\Sigma}_{i}\Phi_{i}(\boldsymbol{o})^{T} + \beta_{i}I.$ (14)

Proof. As mentioned before, w_i and $x_i(o)$ conditioned on w_i are Gaussian distributed, which can be shown as

$$p(\boldsymbol{w}_i) = \mathcal{N}\left(\boldsymbol{w}_i | \hat{\mu}_i, \hat{\Sigma}_i\right),$$

$$p\left(\boldsymbol{x}_i(\boldsymbol{o}) | \boldsymbol{w}_i\right) = \mathcal{N}\left(\boldsymbol{x}_i(\boldsymbol{o}) | \Phi_i(\boldsymbol{o}) \boldsymbol{w}_i, \beta_i I\right).$$

Then, based on the affine transformation property of multivariate Gaussian distributions, the joint distribution of w_i and $x_i(o)$ is given by

$$p(\boldsymbol{w}_i, \boldsymbol{x}_i(\boldsymbol{o})) = \mathcal{N}\left(\begin{pmatrix}\boldsymbol{w}_i\\\boldsymbol{x}_i(\boldsymbol{o})\end{pmatrix}\middle|\begin{pmatrix}\hat{\mu}_i\\\Phi_i(\boldsymbol{o})\hat{\mu}_i\end{pmatrix}, \boldsymbol{\Sigma}_*\right)$$
$$= \mathcal{N}\left(\begin{pmatrix}\boldsymbol{w}_i\\\boldsymbol{x}_i(\boldsymbol{o})\end{pmatrix}\middle|\begin{pmatrix}\hat{\mu}_i\\\hat{\boldsymbol{x}}_i(\boldsymbol{o})\end{pmatrix}, \boldsymbol{\Sigma}_*\right),$$

where

$$\Sigma_* = \begin{pmatrix} \frac{1}{\beta_i} \Phi_i(\boldsymbol{o})^T \Phi_i(\boldsymbol{o}) + (\hat{\Sigma}_i)^{-1} & -\frac{1}{\beta_i} \Phi_i(\boldsymbol{o})^T \\ -\frac{1}{\beta_i} \Phi_i(\boldsymbol{o}) & \frac{1}{\beta_i} \end{pmatrix}^{-1} \\ = \begin{pmatrix} \hat{\Sigma}_i & \hat{\Sigma}_i \Phi_i(\boldsymbol{o})^T \\ \Phi_i(\boldsymbol{o}) \hat{\Sigma}_i & \Phi_i(\boldsymbol{o}) \hat{\Sigma}_i \Phi_i(\boldsymbol{o})^T + \beta_i I \end{pmatrix}.$$



Fig. 3. Data sets that were employed in the simulation experiments. (a) GPS points inside the region. (b) Pols and obstacles inside the region (dots for Pols, red blocks for obstacles).

Thus, we have

$$E\left[(\hat{\boldsymbol{x}}_{i}(\boldsymbol{o}) - \boldsymbol{x}_{i}(\boldsymbol{o}))^{T}(\hat{\boldsymbol{x}}_{i}(\boldsymbol{o}) - \boldsymbol{x}_{i}(\boldsymbol{o}))\right]$$

= $\Phi_{i}(\boldsymbol{o})\hat{\Sigma}_{i}\Phi_{i}(\boldsymbol{o})^{T} + \beta_{i}I,$
 $\forall \boldsymbol{o} \subseteq \mathscr{K} \times \mathscr{P},$

which proves Theorem 3.

PERFORMANCE EVALUATION 6

In this section, we first present detailed experimental settings including the necessary parameters. Next, we compare with four commonly used baselines and discuss the results.

6.1 Setup

Four real data sets are used for the simulation. We employ a taxi mobility traces data set as the participants' trajectories in an MCS campaign, which is collected in Rome, Italy. In the data set, GPS coordinates of approximately 320 taxis are recorded over 30 consecutive days [64]. Each trajectory is marked by a sequence of timestamped GPS points that contain taxi driver ID, timestamp (date and time), and taxi drivers' position (latitude and longitude).

The map offset correction data¹ is used as sensing data contributed by participants. Map offset is a value that indicates the value gap between GPS coordinates in the realworld (i.e., accurate values) and those in a digital map, which is employed as "data quality" in our experiment.

The other two data sets employed for the data calibrating simulation are all air quality monitoring data, one of which is downloaded from OpenSense Zurich Data set [65], and the other one is collected by the Beijing Municipal Environmental Protection testing center, China².

Table 2 summarizes the parameter settings in our experiments. We adopt the following procedures to set up our simulation platform:

For the first data set, which is used as the simulation area for the considered data collection campaign. As all traces are recorded in different parts of Rome. We find an area of about $1000 \times \overline{1000m^2}$ as our

1. (Baidu, Google) map latitude and longitude GPS offset correction. Available: https://www.programering.com/a/MTO1IzNwATg.html 2. http://zx.bjmemc.com.cn/?timestamp=1612083804887

TABLE 2 Parameter of Settings

Parameters	Value
No. of participants	Range from 20% to 100% of the total number of 98 par- ticipants, the default setting is 98
No. of UAVs	Range from 1 to 5, the de- fault setting is 5
Sensing range	Range from 12m to 16m, the default setting is 15m
No. of PoIs	Range from 200 to 300, the default setting is 300
No. of tasks	Range from 1 to 6, the default setting is 6
The amount of budget	Range from 1 200 to 2 200 units, the default setting is 2 000 units
No. of time-slots	17
No. of requested data of each PoI	5
Amount of request reward	Range from 11 to 13 units Randomly

simulation region, Fig. 3(a) shows the GPS points of 30 days inside the region. We randomly select the data recorded in 1 day as locations of participants that perform tasks on the ground. Fig. 3(b) shows the PoIs and obstacles, which are shown as dots and red blocks, respectively.

- We employ the map offset values to indicate a participant's sensing data quality. The map offset of use is nonlinear, in the range of [300, 500] miles. We collect those in the same latitude into a set.
- For the air quality monitoring data set, we employ one subset of data collected by an air quality monitoring station as ground truth, and another subset as participants' sensing data that needs to be calibrated.
- We simulate the UAV as DJI Mavic 2^3 , which in an ideal situation, the maximum speed is 20m/s and the max flight distance is $18\,000m$. The energy cost in this ideal situation includes the necessary signal receiving cost of a UAV. As we described in Section 5.1, the speed and direction of a UAV are decided by the vector acceleration, which is in the range of $[0, 5]m/s^2$.
- The experiments are performed by an Ubuntu 18.04.3 X64 server with an Intel(R) Xeon(R) Gold 5122 CPU (4 cores @3.60Ghz), 62GB memory, and 4 Nvidia GeForce RTX 2080Ti graphics cards. The proposed method is implemented by Python 3.7 and Pytorch 1.7.0. To evaluate the performance of our proposed method, we design the simulation environment based on OpenAI Gym [66], which is a toolkit for developing and comparing reinforcement learning algorithms.
- There are 98 candidates in the selected area who prepare to contribute sensing data. We set the number of UAVs as 5, and the sensing range is 15m. The number of PoIs is 300 with 6 tasks. The total budget is 2000 units. The number of time-slots is set to 17.



Fig. 4. UAV trajectories (lines for UAVs trajectories, red blocks for obstacles, and dots for Pols and participants).

We employ the following four metrics to measure our performance.

• Coverage completed ratio (CCR): The CCR is calculated using Equation (1) to show the ratio between the number of sensed PoIs and the required PoIs of all tasks. The coverage completed ratio is defined as

$$CCR = \sum_{t=1}^{T} \frac{\left| \bigcup_{i \in \{1,2,\dots,I\}} \mathscr{L}_{i}^{h} \right| + \left| \bigcup_{j \in \{1,2,\dots,J\}} \mathscr{L}_{j} \right| * \eta_{p}^{t}}{|\mathscr{P}^{t}| * \eta_{p}^{t} * |\mathscr{K}^{t}|}$$

Calibrating ratio (CR): The CR is calculated as a ratio between the number of effectively calibrated times and the maximum effectively calibrated times Ψ. The maximum calibrated times are decided by experiences in Section 6.2. The calibrating ratio is defined as

ī.

$$CR = \sum_{t=1}^{T} \frac{\left| \bigcup_{j \in \{1, 2, \dots, J\}} \mathscr{L}_{j}^{m} \right|}{\Psi}$$

• Task fairness (TF): The TF is to show how evenly a task associated with PoIs is sensed by participants and UAVs when all tasks are completed. The task fairness is defined as

$$TF = \frac{\left(\sum_{t=1}^{T} \sum_{k=1}^{K^{t}=K} \sum_{p=1}^{P} \hat{\eta}_{p}^{t}(k^{t})\right)^{2}}{\sum_{t=1}^{T} \sum_{p=1}^{P} (\eta_{p}^{t})^{2}}$$

• Energy efficiency (EE): The EE is calculated as a ratio between the number of sensed PoIs and calibration times divided by the energy cost of UAVs. The energy efficiency is defined as

$$EE = \sum_{t=1}^{T} \frac{\left| \bigcup_{j \in \{1,2,\dots,J\}} \mathscr{L}_{j} \right| + \left| \bigcup_{j \in \{1,2,\dots,J\}} \mathscr{L}_{j}^{m} \right|}{\sum_{j=1}^{J} e_{j}(k^{t})}$$

To compare with our proposed algorithm, we first employed a single sensing medium to contribute data, i.e., UAVs or participants, which is referred to as "UAV only" and "Participant only", respectively. Next, we used five baselines to compare with our proposed algorithm. The first one is MADDPG [57], which is a state-of-the-art solution by OpenAI for multi-agent deep reinforcement learning in the competitive and cooperative environment (referred to as "MADDPG"). The state, action, reward function definitions are the same as UMA. In order to allocate tasks jointly to UAVs considering energy cost, the second method [33] transformed the joint optimization problem into a two-sided two-stage matching problem. The method firstly solve the route planning problem based on either dynamic programming or genetic algorithms, then the task assignment problem is addressed by exploring the Gale-Shapley algorithm (referred to as "TARP"). The third method takes an action that maximizes the number of sensed PoIs (referred to as "PoI M"). The fourth one is a greedy approach that navigates a UAV to sense a PoI or meet a participant which could maximize the immediate reward (referred to as "Reward M"). The fifth one allows UAVs to take action randomly (referred to as "Random").

6.2 Simulation Results

We first show moving trajectories for 1, 2, 3, 5 UAVs in Fig. 4. As we described before, there are two responsibilities for UAVs, that is, sensing data from rare sensed PoIs, and calibrating data contributed by participants. As shown in Fig. 4(a) and (b), 1 or 2 UAVs learned to mainly move around in half of the area, responsible for its data collection or calibration, which could potentially maximize their reward. It is worth noting that the blue UAV moved two different trajectories in Fig. 4(a) and (b), as it learned to collaborate with other UAVs. With the increase in the number of UAVs, we observe that trajectories of each UAV are changed. For example, the green one moved a smaller area when a red UAV started to work, as shown in Fig. 4(b) and (c). As we described in Section 5, each UAV has limited observation which is only a part of information of the state space, it has to respond for the limited maximum reward based on the observation. Therefore, the blue and green UAV worked in the upper left and lower right corner, which is shown in Fig. 4(b) and (c), respectively. Furthermore, from Fig. 4(d) we observe that each UAV took responsibility to sense a local region because enough UAVs were deployed and they had learned to collaborate but not to go farther places of other's area. Finally, we see that all UAVs successfully avoid obstacles and never go beyond the border.

The performance results compared with a single sensing medium are shown in Fig. 5. UMA consistently outperforms



Fig. 5. Impact of (a) & (d) number of Pols, (b) number of tasks, (c) sensing range of UAVs, (e) number of UAVs and (f) number of participants on coverage completed ratio, energy efficiency and task fairness.

the other two conditions. For example, in Fig. 5(a) we observe that UMA gains 90.1% more than that of Participant only when the number of PoIs is 200, in terms of coverage completed ratio. Fig. 5(c) shows that UMA gives 13.3% more than that of UAV only when the sensing range of UAVs is 12m, in terms of energy efficiency. We present the time consumption of Algorithm 1 and UMA in Table 3, where Algorithm 1 costs 1.45ms and the UMA consumes 333.43ms when there are 200 PoIs needed to be sensed. Although the values of time consumption rise with the increase of the number of PoIs, the Algorithm 1 and UMA only consume 2.86ms and 336.77ms, when there are 400 PoIs needed to be sensed.

We present the impact of UAV sensing range, budget, number of UAVs, number of PoIs, number of tasks and number of participants on coverage completed ratio, as shown in Fig. 6. Here we fix five parameters described in Section 6.1 and observe the performance of algorithms with the changing of the other one parameter. For example, we fixed the number of UAVs, the total amount of budget, number of PoIs, the number of tasks and the number of participants, while changing the sensing range from 14m to 18m with a step size of 1m (see Fig. 6(a)).

From Fig. 6, we can make the following observations: UMA consistently outperforms all baselines in terms of coverage completed ratio. For example, in Fig. 6(a), we observe that UMA gives 11.0% more than that of TARP and 25.0% more than that of Reward M, when the number of

TABLE 3 Time consumption of Algorithm 1 and UMA

Number of PoIs	100	175	200	225	250	275	300	375	400	1 000	10 000	100 000
Algorithm 1 (ms)	0.77	1.31	1.45	1.59	1.87	2.07	2.16	2.76	2.86	7.19	79.28	950.32
UMA (ms)	327.18	330.20	333.43	332.28	332.75	333.75	333.91	337.57	336.77	358.33	689.47	6048.28



Fig. 6. Impact of (a) number of Pols, (b) number of UAVs, (c) sensing range of UAVs, (d) number of tasks, (e) budget and (f) number of participants on coverage completed ratio

PoIs is 200 and 280, respectively. In Fig. 6(b), we can see that the coverage completed ratio of all methods increases monotonically with the number of UAVs. This is because more UAVs represent better data collection capability. UMA also shows the best performance, e.g., it gives 54.2% more than that of MADDPG. In Fig. 6(c), UMA improves 18.9% of coverage completed ratio if compared to that of PoI M, when the sensing range is 15m. In Fig. 6(d), we observe that UMA improves 58.1% if compared to that of Ramdom, when the number of tasks is 5. Last, in Fig. 6(e), when the total budget of 6 tasks is 1 300 units, UMA gives 18.8% more than that of Reward M. Finally, in Fig. 6(f), UMA achieves a coverage completed ratio of 21.8% if compared to that of PoI M, when there are 60% of total participants.

Next, we present the breakdown results for the other three metrics. Firstly, the impact of the number of UAVs, UAV sensing range, the number of participants, and budget on calibrating ratio are shown in Fig. 7. We observe that UMA outperforms all baselines in terms of calibrating ratio. For example, in Fig. 7(a), we see that calibrating ratio given by UMA rises more intensely than four baselines with a



Fig. 7. Impact of (a) the number of UAVs, (b) sensing range of UAVs, (c) the number of participants, and (d) budget on calibrating ratio.



Fig. 8. Estimation error results after calibrating, where (a) is the experiment with OpenSense Zurich data set, and (b) is the experiment with Beijing air quality monitoring data set.

larger sensing range. For example, UMA improves 56.9% of calibrating ratio if compared to that of PoI M and Reward M, when the number of UAVs is 5. In Fig. 7(b), when the sensing range is 14m, UMA gives a calibrating ratio of 60.7% more if compared to that of PoI M. In Fig. 7(c) and (d), we observe that the calibrating ratio of UMA decreases slightly with more participants and budget.

Two data sets are employed to verify the performance of the proposed system-level calibrating method. Here we employ a method as proposed in [67] to be the compared approach (referred to as "GMR approach"), which uses geometric mean regression to calibrate sensing data. As shown in Fig. 8, with the number of calibrating times increasing from 2 to 8, the estimation errors given by both the proposed and GMR approaches decrease. However, the proposed method performs much better than that of the compared approach. For example, when the number



Fig. 9. Impact of (a) number of UAVs, (b) sensing range of UAVs, (c) number of tasks, and (d) number of Pols on energy efficiency.



Fig. 10. Impact of (a) number of UAVs, (b) sensing range of UAVs, (c) number of tasks, and (d) number of Pols on task fairness.

of calibrating times is 2, the proposed method decreases 69.4% of estimation error compared with that of the GMR approach in Fig 8(a). And the accuracy improves 36.7% on average, compared with that of the GMR approach. On the other side, Fig 8(b) shows that the proposed method decreases 18.2% of estimation error, compared with that of the GMR approach when the number of calibrating times is 4. And the accuracy improves 33.3% on average, compared with that of the GMR approach.

Fig. 9 shows the impact of number of UAVs, sensing range of UAVs, number of tasks, and number of PoIs on energy efficiency. In Fig. 9(a) and (b), We observe that energy is consumed more efficiently with the number of UAVs and sensing range, respectively. While the energy efficiency in Fig. 9(c) and (d) barely changes.

The impact of number of UAVs, sensing range of UAVs,

任务: 1 %Cpu(s): 2 KiB Mem : KiB Swap:	total, 1 9.2 us, 2 65612552 t 2097148 t	running, .6 sy, θ.6 otal, 28974 otal, 1946	0 sleepin) ni, 68.0 1076 free, 5108 free,	ng, 0 stop id, 0.0 wa 11690324 us 151040 us	ped, 0 zomb , 0.0 hi, 6 ed, 24948152 ed. 52833360	bie .2 si, θ.θ st buff/cache avail Mem
进褫USER	PR NI	VIRT	RES SH	HR	EM TIME+	COMMAND
28935 fjh	20	0 18.483g	4.209g 606	6956 R 229.7	6.7 16:20.	56 python3
任务: 1	total, 1	running,	0 sleepin	ig, θstop	oed, θzomb	ie
%Cpu(s): 2	8.6 us, 2	.4 sy, θ.θ	ni, 69.0	id, θ.θ wa	, θ.θ hi, θ	.0 si, 0.0 st
KiB Mem :	6 5612552 t	otal, 28986	528 free,	11677804 use	ed, 24948220	buff/cache
KiB Swap:	2097148 t	otal, 1946	108 free,	151040 use	ed. 52845872	avail Mem
进褫USER	PR NI	VIRT	RES SH	IR	EM TIME+	COMMAND
28935 fjh	20	0 18.483g	4.209g 606	956 R 228.7	6.7 21:42.	99 python3
(5. 欠. 1	totol 1		0	0	and O nomb	
任务: 1 %Cpu(s): 2	total, 1 8 9 us 2	running,	0 sleepin	ng, 0 stop	ped, 0 zomb	Die Dei AAst
任务: 1 %Cpu(s): 2 KiB Mem :	total, 1 8.9 us, 2 65612552 t	running, .5 sy, θ.6	0 sleepin ni, 68.4	ng, 0 stop id, 0.0 wa 11678488 us	ped, 0 zomb , 0.0 hi, 0 ed 24948120	Die D.2 si, θ.θ st buff/cache
任务: 1 %Cpu(s): 2 KiB Mem : KiB Swap:	total, 1 8.9 us, 2 65612552 t 2097148 t	l running, 2.5 sy, 0.6 otal, 28985 otal, 1946	0 sleepin) ni, 68.4 5944 free, 5108 free,	ng, 0 stop id, 0.0 wa 11678488 us 151040 us	ped, 0 zomb , 0.0 hi, 0 ed, 24948120 ed. 52845196	die .2 si, 0.0 st buff/cache avail Mem
任务: 1 %Cpu(s): 2 KiB Mem : KiB Swap: 进瘾USER	total, 1 8.9 us, 2 65612552 t 2097148 t	running, 2.5 sy, θ.6 otal, 28985 otal, 1946	0 sleepin) ni, 68.4 5944 free, 5108 free, RES SH	ng, 0 stop id, 0.0 wa 11678488 us 151040 us R #%ΩPU %M	ped, 0 zomb , 0.0 hi, 0 ed, 24948120 ed. 52845196	ie .2 si, 0.0 st buff/cache avail Mem COMMAND
任务: 1 %Cpu(s): 2 KiB Mem : KiB Swap: 进稿USER 28935 fjh	total, 1 8.9 us, 2 65612552 t 2097148 t PR NI 20	e running, 2.5 sy, θ.6 cotal, 28985 cotal, 1946 VIRT θ 18.483g	0 sleepin 0 ni, 68.4 5944 free, 5108 free, RES SH 4.209g 600	ng, 0 stop id, 0.0 wa 11678488 us 151040 us IR 走修CPU %M 5956 R 225.7	ped, 0 zomb , 0.0 hi, 0 ed, 24948120 ed. 52845196 EM TIME+ 6.7 27:53.	vie 1.2 si, 0.0 st buff/cache avail Mem <u>COMMAND</u> 99 python3
任务: 1 %Cpu(s): 2 KiB Mem : KiB Swap: 进褫USER 28935 fjh 任条: 1	total, 1 8.9 us, 2 65612552 t 2097148 t PR NI 20 total, 1	running, 2.5 sy, θ.6 cotal, 28985 cotal, 1946 VIRT θ 18.483g running.	θ sleepin 0 ni, 68.4 5944 free, 6108 free, RES SH 4.209g 606 θ sleepin	ng, 0 stopp id, 0.0 wa 11678488 us 151040 us R 走後CPU %M 5956 R 225.7	ped, θ zomb , 0.0 hi, θ ed, 24948120 ed. 52845196 EM TIME+ 6.7 27:53.	vie 1.2 si, 0.0 st buff/cache avail Mem COMMAND 99 python3 vie
任务: 1 %Cpu(s): 2 KiB Mem : KiB Swap: 进稿USER 28935 fjh 任务: 1 %Cpu(s): 2	total, 1 8.9 us, 2 65612552 t 2097148 t <u>PR NI</u> 20 total, 1 8.7 us, 2	running, .5 sy, θ.6 total, 28985 total, 1946 <u>VIRT</u> θ 18.483g running, .5 sy, θ.6	0 sleepin 0 ni, 68.4 5944 free, 5108 free, RES SF 4.209g 606 0 sleepin ni, 68.5	ng, 0 stop id, 0.0 wa 11678488 us 151040 us R 走豫CPU %M 5956 R 225.7 ng, 0 stop id, 0.0 wa	ped, 0 zomb , 0.0 hi, 0 ed, 24948120 ed. 52845196 EM TIME+ 6.7 27:53. ped, 0 zomb , 0.0 hi, 0	ie .2 si, 0.0 st buff/cache avail Mem COMMAND 99 python3 vie .3 si, 0.0 st
任务: 1 %Cpu(s): 2 KiB Mem : KiB Swap: 进続USER 28935 fjh 任务: 1 %Cpu(s): 2 KiB Mem :	total, 1 8.9 us, 2 65612552 t 2097148 t <u>PR NI</u> 20 total, 1 8.7 us, 2 65612552 t	. running, .5 sy, θ.6 cotal, 28985 cotal, 1946 VIRT θ 18.483g . running, .5 sy, θ.6 cotal, 29031	<pre>0 sleepin) ni, 68.4 ;944 free, ;108 free, RES SH 4.209g 606 0 sleepin) ni, 68.5 ;1652 free,</pre>	ng, 0 stop id, 0.0 wa 11678488 us 151040 us R 走後CPU %M 5956 R 225.7 ng, 0 stop id, 0.0 wa 11632644 us	ped, θ zomb , θ.θ hi, θ ed, 24948120 ed. ed. 52845196 TIME+ 6.7 27:53. ped, θ zomb ped, 0.0 hi, θ comb	<pre>bie buff/cache avail Mem COMMAND 99 python3 bie .3 si, 0.0 st buff/cache</pre>
任务: 1 %Cpu(s): 2 KiB Mem : HiB Swap: 进稿[USER 28935 fjh 任务: 1 %Cpu(s): 2 KiB Mem : KiB Swap:	total, 1 8.9 us, 2 65612552 t 2097148 t PR NI 20 total, 1 8.7 us, 2 65612552 t 2097148 t	. running, 2.5 sy, θ.6 cotal, 28985 cotal, 1946 VIRT θ 18.483g . running, 2.5 sy, θ.6 cotal, 29033 cotal, 1946	 θ sleepin ni, 68.4 6944 free, 6968 free, RES SH 4.269g 666 θ sleepin ni, 68.5 1652 free, 1688 free, 	ng, 0 stop id, 0.0 wa 11678488 us 151040 us iR € %CPU %M i956 R 225.7 ng, 0 stop id, 0.0 wa 11632644 us 151040 us	ped, 0 zomb , 0.0 hi, 0 ed, 24948120 ed. 52845196 EM TIME+ 6.7 27:53. ped, 0 zomb , 0.0 hi, 0 ed, 24948256 ed. 52891132	bie 9.2 si, 0.0 st buff/cache avail Mem COMMAND 99 python3 bie .3 si, 0.0 st buff/cache avail Mem
任务: 1 %Cpu(s): 2 KiB Mem : KiB Swap: 进稿USER 28935 fjh 任务: 1 %Cpu(s): 2 KiB Mem : KiB Swap: 进褫USER	total, 1 8.9 us, 2 65512552 t 2097148 t PR NI 20 total, 1 8.7 us, 2 65612552 t 2097148 t PR NI	 running, 5 sy, 0.6 cotal, 28985 cotal, 28985 cotal, 1946 VIRT 0 18.483g running, .5 sy, 0.6 cotal, 29031 cotal, 1946 VIRT 	<pre>0 sleepin) ni, 68.4)944 free, j108 free, RES SH 4.209g 606 0 sleepin) ni, 68.5 1652 free, j108 free, RES SH</pre>	ng, 0 stop id, 0.0 wa 11678488 us 151640 us is 2000 wa is 2000 wa	ped, 0 zomti , 0.0 hi, 0 ed, 24948120 0 ed. 52845196 1 EM TIME+ 6.7 27:53. ped, 0.2 omti 6, 0 6, 0 ped, 0.0 hi, 0 6 , 2,24948256 52891132 0 ed. 52891132 TIME+ 1	<pre>bie buff/cache avail Mem COMMAND 99 python3 bie buff/cache avail Mem COMMAND</pre>

Fig. 11. Resource utilization of CPU and memory for performing the proposed method.

number of tasks, and number of PoIs on task fairness is shown in Fig. 10. Similar to Fig. 9(a), task fairness of UMA increases monotonically with the number of UAVs. And the task fairness increases slightly with the number of participants and PoIs. It is worth noting that, with the number of tasks increasing, task fairness decreases. However, we observe that UMA still outperforms all baselines in terms of task fairness.

Finally, we record the resource utilization for performing the proposed method UMA under the condition of 5 UAVs, 2000 units budget and 300 PoIs. As shown in Fig. 11, the range of CPU utilization is between 225.7% and 237.7% where the total utilization is 400% when the 4 cores are fully utilized. The memory utilization is stable at 6.7%.

7 CONCLUSION AND FUTURE WORK

In this paper, we explicitly consider the problem of UAV-assisted multi-task allocation for MCS to maximize sensing coverage. To deal with the problem, we proposed a novel method "UMA". On one hand, the method incentivized participants to contribute high quality sensing data, with a limited budget. On the other hand, the UAVs were employed to sense data from rarely sensed PoIs. In the meanwhile, they were also employed to calibrate for sensors of participants. The results well justified the efficiency and robustness of UMA in terms of four metrics, coverage completed ratio, calibrating ratio, task fairness and energy efficiency, compared with the state-of-the-art.

In the future, we plan to propose a method that determines the number of PoIs and pieces of sensing data, to mitigate data collection redundancy. Besides the budget and maximum offered reward, the task quality requirement is also considered to calculate the number of pieces of data to be sensed from a PoI. Normally higher quality sensing data requires more sensing data. We attempt to leverage the confidence interval to quantify the sensing data quality requirement. In practice, when the confidence interval is short, more data should be collected from the PoI in question. A reinforcement learning method may be employed to find out the relationship between the value of the confidence interval and the amount of the required sensing data.

REFERENCES

- A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE communications surveys* & tutorials, vol. 21, no. 3, pp. 2419–2465, 2019.
- [2] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. S. Shen, "Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1317–1331, 2019.
- [3] X. Li and X. Zhang, "Multi-task allocation under time constraints in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1494–1510, 2021.
- [4] B. Zhao, S. Tang, X. Liu, and X. Zhang, "Pace: privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1924–1939, 2021.
- [5] X. Chen, S. Xu, J. Han, H. Fu, X. Pi, C. Joe-Wong, Y. Li, L. Zhang, H. Y. Noh, and P. Zhang, "Pas: Prediction-based actuation system for city-scale ridesharing vehicular mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3719–3734, 2020.
- [6] K. Lou, Y. Yang, E. Wang, Z. Liu, T. Baker, and A. K. Bashir, "Reinforcement learning based advertising strategy using crowdsensing vehicular data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4635–4647, 2021.
- [7] M. Noreikis, Y. Xiao, J. Hu, and Y. Chen, "Snaptask: Towards efficient visual crowdsourcing for indoor mapping," in *Proc. IEEE ICDCS 2018*. IEEE, 2018, pp. 578–588.
 [8] A. Hamrouni, H. Ghazzai, M. Frikha, and Y. Massoud, "A spa-
- [8] A. Hamrouni, H. Ghazzai, M. Frikha, and Y. Massoud, "A spatial mobile crowdsourcing framework for event reporting," *IEEE transactions on computational social systems*, vol. 7, no. 2, pp. 477– 491, 2020.
- [9] X. Wang, Z. Ning, X. Hu, E. C.-H. Ngai, L. Wang, B. Hu, and R. Y. Kwok, "A city-wide real-time traffic management system: Enabling crowdsensing in social internet of vehicles," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 19–25, 2018.
- [10] L. Wang, Z. Yu, D. Zhang, B. Guo, and C. H. Liu, "Heterogeneous multi-task assignment in mobile crowdsensing using spatiotemporal correlation," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 84–97, 2018.
- [11] F. Restuccia, N. Ghosh, S. Bhattacharjee, S. K. Das, and T. Melodia, "Quality of information in mobile crowdsensing: Survey and research challenges," ACM Transactions on Sensor Networks, vol. 13, no. 4, pp. 1–43, 2017.
- [12] D. Wu, T. Xiao, X. Liao, J. Luo, C. Wu, S. Zhang, Y. Li, and Y. Guo, "When sharing economy meets IoT: Towards fine-grained urban air quality monitoring through mobile crowdsensing on bikeshare system," *Proc. ACM IMWUT 2020*, vol. 4, no. 2, pp. 1–26, 2020.
- [13] S. Bhattacharjee, N. Ghosh, V. K. Shah, and S. K. Das, "QnQ: Quality and quantity based unified approach for secure and trustworthy mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 1, pp. 200–216, 2018.
- [14] X. Gong and N. B. Shroff, "Truthful mobile crowdsensing for strategic users with private data quality," *IEEE/ACM Transactions* on Networking, vol. 27, no. 5, pp. 1959–1972, 2019.
- [15] J. Wang, Y. Wang, D. Zhang, F. Wang, H. Xiong, C. Chen, Q. Lv, and Z. Qiu, "Multi-task allocation in mobile crowd sensing with individual task quality assurance," *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2101–2113, 2018.
- [16] C. Zhao, S. Yang, P. Yan, Q. Yang, X. Yang, and J. McCann, "Data quality guarantee for credible caching device selection in mobile crowdsensing systems," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 58–64, 2018.
- [17] N. B. Truong, G. M. Lee, T.-W. Um, and M. Mackay, "Trust evaluation mechanism for user recruitment in mobile crowd-sensing in the internet of things," *IEEE Transactions on Information Forensics* and Security, vol. 14, no. 10, pp. 2705–2719, 2019.

- [18] D. Peng, F. Wu, and G. Chen, "Data quality guided incentive mechanism design for crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 307–319, 2018.
- [19] S. Liu, Z. Zheng, F. Wu, S. Tang, and G. Chen, "Context-aware data quality estimation in mobile crowdsensing," in *Proc. IEEE INFOCOM 2017.* IEEE, 2017, pp. 1–9.
- [20] Y. Li, H. Xiao, Z. Qin, C. Miao, L. Su, J. Gao, K. Ren, and B. Ding, "Towards differentially private truth discovery for crowd sensing systems," in *Proc. IEEE ICDCS 2020*. IEEE, 2020, pp. 1156–1166.
- [21] R. Tan, G. Xing, Z. Yuan, X. Liu, and J. Yao, "System-level calibration for data fusion in wireless sensor networks," ACM *Transactions on Sensor Networks*, vol. 9, no. 3, pp. 1–27, 2013.
- [22] K. Fu, W. Ren, and W. Dong, "Multihop calibration for mobile sensing: K-hop calibratability and reference sensor deployment," in *Proc. IEEE INFOCOM 2017*. IEEE, 2017, pp. 1–9.
- [23] H. Gao, C. H. Liu, J. Tan, D. Yang, P. Hui, and W. Wang, "Online quality-aware incentive mechanism for mobile crowd sensing with extra bonus," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2589–2603, 2019.
- [24] C. H. Liu, Z. Dai, Y. Zhao, J. Crowcroft, D. Wu, and K. K. Leung, "Distributed and energy-efficient mobile crowdsensing with charging stations by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 130–146, 2021.
- [25] L. Wang, Z. Yu, D. Zhang, B. Guo, and C. H. Liu, "Heterogeneous multi-task assignment in mobile crowdsensing using spatiotemporal correlation," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 84–97, 2019.
- [26] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu, "Truthful incentive mechanism for nondeterministic crowdsensing with vehicles," *IEEE Transactions on Mobile Computing*, vol. 17, no. 12, pp. 2982– 2997, 2018.
- [27] C. H. Liu, Z. Chen, and Y. Zhan, "Energy-efficient distributed mobile crowd sensing: A deep learning approach," *IEEE Journal* on Selected Areas in Communications, vol. 37, no. 6, pp. 1262–1276, 2019.
- [28] H. Wu, X. Tao, N. Zhang, and X. Shen, "Cooperative UAV clusterassisted terrestrial cellular networks for ubiquitous coverage," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2045–2058, 2018.
- [29] C. H. Liu, X. Ma, X. Gao, and J. Tang, "Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1274–1285, 2019.
- [30] D. Wu, T. Xiao, X. Liao, J. Luo, C. Wu, S. Zhang, Y. Li, and Y. Guo, "When sharing economy meets iot: Towards fine-grained urban air quality monitoring through mobile crowdsensing on bike-share system," *Proc. ACM IMWUT 2020*, vol. 4, no. 2, pp. 1–26, 2020.
- [31] P. Huang, X. Zhang, L. Guo, and M. Li, "Incentivizing crowdsensing-based noise monitoring with differentially-private locations," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 519–532, 2021.
- [32] X. Wang, J. Zhang, X. Tian, X. Gan, Y. Guan, and X. Wang, "Crowdsensing-based consensus incident report for road traffic acquisition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2536–2547, 2018.
- [33] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, and M. Guizani, "When mobile crowd sensing meets UAV: Energyefficient task assignment and route planning," *IEEE Transactions* on Communications, vol. 66, no. 11, pp. 5526–5538, 2018.
- [34] X. Li and X. Zhang, "Multi-task allocation under time constraints in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1494–1510, 2021.
- [35] H. Zhao, M. Xiao, J. Wu, Y. Xu, H. Huang, and S. Zhang, "Differentially private unknown worker recruitment for mobile crowdsensing using multi-armed bandits," *IEEE Transactions on Mobile Computing*, vol. 20, no. 9, pp. 2779–2794, 2021.
- [36] Z. Cai, Z. Duan, and W. Li, "Exploiting multi-dimensional task diversity in distributed auctions for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 8, pp. 2576–2591, 2021.
- [37] F. Wu, S. Yang, Z. Zheng, S. Tang, and G. Chen, "Fine grained user profiling for personalized task matching in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 10, pp. 2961– 2976, 2021.
- [38] S. Yang, K. Han, Z. Zheng, S. Tang, and F. Wu, "Towards personalized task matching in mobile crowdsensing via fine-grained user profiling," in *Proc. IEEE INFOCOM 2018*, 2018, pp. 2411–2419.

- [39] Z. Wang, J. Zhao, J. Hu, T. Zhu, Q. Wang, J. Ren, and C. Li, "Towards personalized task-oriented worker recruitment in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 2080–2093, 2021.
- [40] G. Yang, B. Wang, X. He, J. Wang, and H. Pervaiz, "Competitioncongestion-aware stable worker-task matching in mobile crowd sensing," *IEEE Transactions on Network and Service Management*, early access, doi: 10.1109/TNSM.2021.3072638.
- [41] J. Wang, F. Wang, Y. Wang, D. Zhang, L. Wang, and Z. Qiu, "Socialnetwork-assisted worker recruitment in mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1661– 1673, 2019.
- [42] F. Yucel, M. Yuksel, and E. Bulut, "QoS-based budget constrained stable task assignment in mobile crowdsensing," *IEEE Transactions* on Mobile Computing, early access, doi: 10.1109/TMC.2020.2997280.
- [43] G. Yang, Y. Li, X. He, Y. Song, J. Wang, and M. Liu, "Profilefree and real-time task recommendation in mobile crowdsensing," *IEEE Transactions on Computational Social Systems*, early access, doi: 10.1109/TCSS.2021.3073031.
- [44] C. Dai, X. Wang, K. Liu, D. Qi, W. Lin, and P. Zhou, "Stable task assignment for mobile crowdsensing with budget constraint," *IEEE Transactions on Mobile Computing*, early access, doi: 10.1109/TMC.2020.3000234.
- [45] F. Restuccia, P. Ferraro, S. Silvestri, S. K. Das, and G. L. Re, "Incentme: effective mechanism design to stimulate crowdsensing participants with uncertain mobility," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1571–1584, 2018.
- [46] S. Xu, X. Chen, X. Pi, C. Joe-Wong, P. Zhang, and H. Y. Noh, "ilocus: Incentivizing vehicle mobility to optimize sensing distribution in crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 8, pp. 1831–1847, 2019.
- [47] G. Fan, H. Jin, Q. Liu, W. Qin, X. Gan, H. Long, L. Fu, and X. Wang, "Joint scheduling and incentive mechanism for spatio-temporal vehicular crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1449–1464, 2021.
- [48] J. Hu, K. Yang, K. Wang, and K. Zhang, "A blockchain-based reward mechanism for mobile crowdsensing," *IEEE Transactions* on Computational Social Systems, vol. 7, no. 1, pp. 178–191, 2020.
- [49] J. Nie, J. Luo, Z. Xiong, D. Niyato, P. Wang, and H. V. Poor, "A multi-leader multi-follower game-based analysis for incentive mechanisms in socially-aware mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1457–1471, 2021.
- [50] Y. Zhang, X. Zhang, and F. Li, "Bicrowd: Online biobjective incentive mechanism for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 11078–11091, 2020.
- [51] Z. Zheng, S. Yang, J. Xie, F. Wu, X. Gao, and G. Chen, "On designing strategy-proof budget feasible online mechanism for mobile crowdsensing with time-discounting values," *IEEE Transactions on Mobile Computing*, early access, doi:10.1109/TMC.2020.3034499.
- [52] X. Zhu, Y. Luo, A. Liu, W. Tang, and M. Z. A. Bhuiyan, "A deep learning-based mobile crowdsensing scheme by predicting vehicle mobility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4648–4659, 2021.
- [53] X. Tao and A. S. Hafid, "DeepSensing: A Novel Mobile Crowdsensing Framework With Double Deep Q-Network and Prioritized Experience Replay," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11547–11558, 2020.
- [54] Y. Hu, J. Wang, B. Wu, and S. Helal, "Rl-recruiter+: Mobilitypredictability-aware participant selection learning for fromscratch mobile crowdsensing," *IEEE Transactions on Mobile Computing*, early access, doi: 10.1109/TMC.2021.3077636.
- [55] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., "Human-level control through deep reinforcement learning," nature, vol. 518, no. 7540, pp. 529–533, 2015.
- [56] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.
- [57] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in neural information processing systems*, 2017, pp. 6379–6390.
- [58] P. Ferrer-Cid, J. M. Barcelo-Ordinas, J. Garcia-Vidal, A. Ripoll, and M. Viana, "Multisensor data fusion calibration in IoT air pollution platforms," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3124– 3132, 2020.

- [59] Y. Lin, W. Dong, and Y. Chen, "Calibrating low-cost sensors by a two-phase learning approach for urban air quality measurement," *Proc. ACM IMWUT 2018*, vol. 2, no. 1, pp. 1–18, 2018.
- [60] H. Gao, Y. Xiao, H. Yan, Y. Tian, D. Wang, and W. Wang, "A learning-based credible participant recruitment strategy for mobile crowd sensing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5302–5314, 2020.
- [61] Z. Li, T. Xi, J. Deng, G. Zhang, S. Wen, and R. He, "Gp-nas: Gaussian process based neural architecture search," in *Proc. IEEE/CVF CVPR* 2020, 2020, pp. 11933–11942.
- [62] D. Zachariah and P. Stoica, "Cramer-rao bound analog of bayes' rule [lecture notes]," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 164–168, 2015.
- [63] J. V. Candy, Bayesian signal processing: classical, modern, and particle filtering methods. John Wiley & Sons, 2016, vol. 54.
- [64] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, "CRAWDAD dataset roma/taxi (v. 2014-07-17)," Downloaded from http://crawdad.org/roma/taxi/20140717, Jul. 2014.
- [65] J. J. Li, B. Faltings, O. Saukh, D. Hasenfratz, and J. Beutel, "Sensing the air we breathe—the opensense zurich dataset," in *Proc. AAAI* 2012, vol. 26, no. 1, 2012.
- [66] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," arXiv preprint arXiv:1606.01540, 2016.
- [67] O. Saukh, D. Hasenfratz, and L. Thiele, "Reducing multi-hop calibration errors in large-scale mobile sensor networks," in *Proc.* ACM IPSN 2015. ACM, 2015, pp. 274–285.



Hui Gao received the master's and Ph.D. degrees from Beijing University of Posts and Telecommunications, China, in 2013 and 2017. He is currently an Assistant Professor in the School of Computer Science (National Pilot Software Engineering School) at Beijing University of Posts and Telecommunications. His research interests include mobile crowdsensing, deep learning for smart city, and Internet-of-Things.



Jianhao Feng received the bachelor's degree in the School of Computer Science (National Pilot Software Engineering School) from Beijing University of Posts and Telecommunications, where he is currently pursuing the M.S. degree. His research interests include mobile crowdsensing, reinforcement learning and motion retargeting.



Yu Xiao received the bachelor's and master's degrees in computer science and technology from the Beijing University of Posts and Telecommunications, China, and the Ph.D. degree (Hons.) in computer science from Aalto University, Espoo, Finland, in January 2012. She is currently an Assistant Professor with the Department of Communications and Networking, Aalto University, where she leads the Mobile Cloud Computing Group. Her research interests include edge computing, mobile crowdsensing,

and energy-efficient wireless networking. Dr. Xiao has received three best paper awards from IEEE/ACM Conferences. She is also a recipient of the three-year postdoc grant from the Academy of Finland.

IEEE TRANS. ON MOBILE COMPUTING, VOL. XX, NO. XX, XX XX



Bo Zhang received the master's and Ph.D. degrees from the Beijing University of Posts and Telecommunications, China, in 2010 and 2016, where he is currently an Assistant Professor with the State Key Laboratory of Networking and Switching Technology. His research interests include mobile crowd sensing, Internet of Things, deep learning for smart city, medical image analysis, and its applications.



Wendong Wang received the bachelor's and master's degrees from Beijing University of Posts and Telecommunications in 1985 and 1991, respectively, where he is currently a Full Professor in State Key Laboratory of Networking and Switching Technology. He has published over 200 papers in various journals and conference proceedings. His current research interests are the next generation network architecture and mobile crowdsensing. He is a member of IEEE.