A Coverage-aware Task Allocation Method for UAV-assisted Mobile Crowd Sensing

Xinbin Liu, Ye Wang, Hui Gao, Edith C.H. Ngai, *Member, IEEE,* Bo Zhang, *Member, IEEE,* Chuhan Wang, and Wendong Wang, *Member, IEEE*

***Abstract*—Mobile Crowd Sensing (MCS) is an emerging paradigm that engages participants collaboratively in completing sensing tasks. The mobility and intelligence of mobile devices offer an efficient solution for large-scale sensing applications, such as in smart cities. Unmanned aerial vehicles (UAVs), considered as mobile devices, can be integrated into MCS to collaborate with human participants in order to meet the task sensing coverage requirement. In this paper, we investigate a UAV-assisted task allocation method (U-TAM) that allocates tasks to human participants and UAVs concurrently. Distinct from existing methods, U-TAM prioritizes minimizing the privacy leakage of human participants while maximizing sensed coverage. To achieve this, it initially predicts their trajectories using a deep reinforcement learning approach, relying solely on the information provided by their start and destination locations. In addition to the predicted trajectories, the proposed U-TAM allocates tasks to human participants based on their tolerance levels and limited budget. This approach aligns with the Pareto optimal theory, seeking to balance the trade-off among participants' tolerance level, limited budget, and the requirement for task sensing coverage. In the meantime, the UAVs sense data efficiently from areas that are not sensed by human participants or other UAVs. To this end, we propose a multi-agent deep reinforcement learning framework for multi-UAV trajectory planning, which integrates the greedy method into deep Q-learning. We evaluate the proposed method using simulation and a small-scale practical experiment. Extensive experiments are used to verify the method's efficiency.**

***Index Terms*—Mobile crowd sensing, task allocation, Pareto optimal theory, UAV assistance, reinforcement learning.**

I. Introduction

Mobile crowd sensing (MCS) is a common sensing paradigm that may assemble a crowd of mobile users to carry out a range of sensing activities as a result of the development of smart mobile devices with strong sensing, networking, and computing capabilities [1]–[4]. These benefits have made a wide range of MCS applications possible, including environment monitoring [5], public event reporting [6], road and traffic monitoring [7].

One of the crucial aspects of the MCS system, vital for ensuring the quality of sensing data, lies in the task allocation process, which aims to meet sensing coverage requirements within limited resources, such as budget constraints [8]. Sensing coverage holds paramount significance in both spatial and temporal dimensions [9], [10]. In practice, self-planned paths by human participants may not fully address the platform's spatial-temporal coverage needs [11]. Additionally, existing task allocation approaches primarily concentrate on selecting a group of human participants under a limited budget without considering their constraints, which indicate the number of pieces of sensing data they intend to contribute. In reality, a participant may not be able to contribute over-many pieces of sensing data for MCS tasks, e.g., when he/she is steering or the battery life of the mobile device is short [12]. Finding a trade-off between the quantity of sensing data to be experienced and a human participant's constraint is therefore crucial. For example, Xu *et al.* addressed the constrained sensing distance of human participants by planning a path for each participant within this limited range [13]. In another approach, Huang *et al.* proposed a method that allocated tasks to human participants based on their available time without considering task coverage requirements [14]. In practice, human participants are not full-time employees dedicated to performing sensing tasks, making it challenging to mandate data collection from specific locations. Additionally, the allocation of tasks to human participants to meet coverage requirements warrants consideration.

For the establishment of ubiquitous MCS services, the unmanned aerial vehicle (UAV), distinguished by its high agility and flexibility, is increasingly recognized as a powerful assistant for MCS. It facilitates the collection of sensing data in challenging, hard-to-reach, and infrastructure-constrained areas. Wang *et al.* and Jiang *et al.* proposed a UAV crowd-sensing system that employed UAVs to perform tasks [15], [16]. However, due to restrictions in certain cities, such as Beijing, Seoul, and Washington D.C., where the use of UAVs is prohibited, human participants remain essential to ensure the continuity of MCS campaigns. Samir *et al.* proposed a UAV-assisted vehicular network wherein sensors on vehicles generate data streams, and UAVs were used to collect and process this data [17]. In [18], the trajectory of multiple unmanned

This work was partly supported by the National Natural Science Foundation of China (Grant No. 62002025, 62072047), the UGC General Research Fund (Grant No. 17203320, 17209822), and the National Key Research and Development Program of China (Grant NO. 2023YFB2904103).

Xinbin Liu, Ye Wang and Hui Gao (Corresponding author) is with School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China (E-mail: {code, wye, gaohui786}@bupt.edu.cn).

Edith C.H. Ngai (Corresponding author) is with Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong, China (E-mail: chngai@eee.hku.hk).

Bo Zhang and Wendong Wang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (E-mail: {zbo, wdwang}@bupt.edu.cn).

Chuhan Wang is with School of Physics and Optoelectronic Engineering, Guangdong University of Technology, Guangzhou, China (E-mail: 3220007252@mail2.gdut.edu.cn).

Copyright © 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

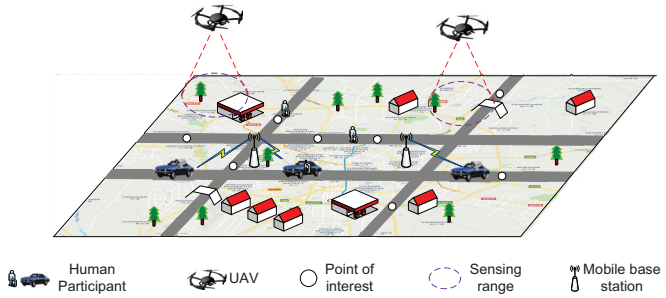

Fig. 1. The proposed hybrid MCS scenario.

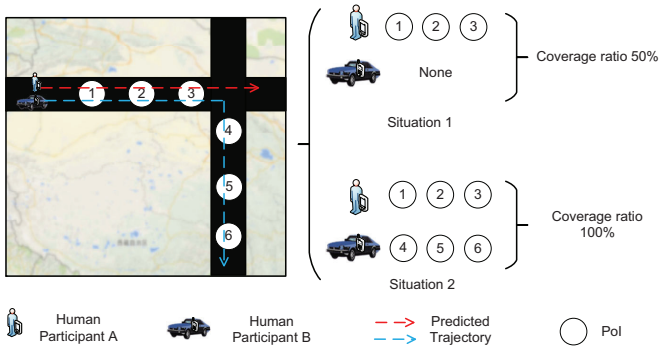

Fig. 2. A task allocation example. The coverage ratio is 50% in situation 1, while there is a chance to be fully covered in situation 2.

ground vehicles (UGVs) and UAVs was jointly considered to relay information for users. Both authors proposed scenarios in which vehicles or humans collect data, and UAVs are employed to relay or process the data. In reality, obstacles such as buildings need to be considered when planning UAV trajectories [19].

The hybrid MCS scenario is illustrated in Fig. 1, akin to the approach presented in [20], [21]. Assuming there are MCS tasks requiring the utilization of sensing information in a specific region, such as noise level [22] and air quality [23], various points of interest (PoIs) are scattered throughout the designated area, requiring perception by either human participants or UAVs. Human participants may opt to apply for a task based on their requested reward and interest in participating. Concurrently, the UAVs, serving as an auxiliary component, initiate their cruising phase to provide sensing data.

The endurance and performance of UAVs are limited by their on-board battery capacity. Energy efficiency emerges as a crucial prerequisite for these hybrid MCS systems. Put differently, to avert energy wastage, careful arrangement of each UAV's trajectory, considering the positions of human participants, other UAVs, and PoIs, becomes imperative. For the human participants, a balance should be considered between their tolerance levels and task sensing coverage requirement. We take Fig.2 as an example, consider a scenario where human participant A can contribute no more than three pieces of sensing data. If PoIs 1, 2, and 3 are allocated to A based on the predicted trajectory, there would be no other human participant available to cover PoIs 1, 2, and 3. However, if both human participants A and B cover PoIs 1, 2, 3, and 4, 5, 6, respectively, the sensing coverage requirement will be fulfilled. Present approaches frequently address these issues independently, lacking a comprehensive investigation from the standpoint of human-UAV cooperation regarding work distribution.

In order to address the issue, we suggest in this study a UAV-assisted task allocation method (U-TAM) for MCS systems to satisfy the task sensing coverage requirement. Specifically, the proposed U-TAM takes task assignment optimization into account. For the human participants, to cover the sensing region and mitigate potential leakage of personal trajectory information, U-TAM initially employs a reinforcement learning method to predict participant trajectories. Subsequently, it allocates tasks based on the Pareto optimal theory, considering the predicted routes and tolerance levels of participants. On the other side, for the UAVs, the U-TAM plans their trajectories to sense data while taking into account the locations of human participants, other UAVs, obstacles, and PoIs that human participants do not frequently visit. In this context, the U-TAM utilizes a multi-agent deep reinforcement learning framework to plan trajectories for multiple UAVs, integrating the greedy method into deep Q-learning.

The main contribution of this paper is summarized in the following:

- To maximize sensing coverage, we introduce a UAV-assisted task allocation method that concurrently assigns tasks to both human participants and UAVs. We formulate the optimization problem as NP-hard.
- Specifically, the proposed method takes into account both human participants' tolerance levels and UAVs' sensing efficiency. On one hand, the method allocates a limited number of tasks to human participants under tolerance constraints. On the other hand, the approach plans a sensing path for each UAV, considering the locations of participants, other UAVs, obstacles, and PoIs.
- We verify the effectiveness and efficiency of the proposed U-TAM by conducting a real experiment and simulation over real datasets.

The rest of this paper is organized as follows. We discuss related research efforts in Section II. The system model is described in Section III. We propose the trajectory-based task allocation method for human participants in Section IV. Last, the UAV trajectory scheduling algorithm is introduced in Section V. Finally, we present the simulation results in Section VI and conclude the paper in Section VII.

II. Related Work

In this section, we study the pertinent literature for two topics: task allocation strategies for MCS and UAV-assisted MCS.

A. Task allocation methods

Zhu *et al.* proposed a shared bicycle return framework that allocated shared bike return station tasks to participants who rented the shared bikes, in order to re-balancing the number

TABLE I
COMPARISON OF RELATED WORKS WITH OUR WORK

References	Hybrid sensing scenario	Multi-UAV	Obstacle considered	Optimization objective	Optimization method
[17]	✓	✓	✗	Sum age of information minimization	DDPG
[18]	✓	✓	✗	Average spectrum efficiency maximization	DDPG
[19]	✗	✓	✓	Collect data quantity maximization	DRL
[24]	✗	✓	✓	Collect data quantity maximization	D-DQN
[25]	✗	✗	✓	Average age of information minimization	SAC+AO
[26]	✗	✓	✗	Delivery reward maximization	DDPG
[27]	✗	✓	✓	Data collection time minimization	DRL
[28]	✗	✓	✗	Average age of information minimization	DRL
Our work	✓	✓	✓	Data coverage maximization	DQN+GREEDY

of shared bikes of each station. The framework employed a deep neural network to forecast the trends of bike demand hot spots, and a reinforcement learning method was used to allocate tasks [29]. Zhao *et al.* proposed a privacy-preserving system that allocated tasks to participants who claimed a low requested reward [30]. Ding *et al.* in [31] considered the time-sensitive task allocation problem, the method first computed the probabilities of participant reaching the data sensed area before the deadline of the task, then allocated the task to participant using the semi-Markov prediction and meta-path prediction, respectively. Wang *et al.* proposed a task allocation method in which tasks were allocated automatically to participants in order to maximize the sensing quality [32]. Zhang *et al.* proposed a multi-task allocation method that also took participant privacy protection into consideration. With the purpose of enhancing the human participant and task completion rate, authors evaluated the user reputation levels, preferences, and task attributes utilizing the privacy-preserving grouping and matching mechanisms [33]. Xiao *et al.* proposed a participant selection method to alleviate the problem of participants contributing low quality data. The method required participants to upload their locations and expertise first, and used an incentive mechanism to motivate the selected participants to contribute high quality data [34]. Xu *et al.* in [35] focused on allocating tasks to enough participants. The method allowed the selected participants to recruit their social friends. Huang *et al.* in [36] proposed a two-stage task allocation method. In the first stage, the method allocated tasks to participants with the goal of maximizing the task publisher's revenue. For the second stage, the available participants who were not recruited in the first stage still had a chance to be assigned tasks. In order to minimize participants' travel detour costs, Wang *et al.* proposed a task allocation method that constructed an assignment graph to model the assignment relationship between the tasks and participants. Then the method assigned tasks to participants to achieve Pareto-optimal schemes [37]. Yin *et al.* proposed a task allocation framework that considered the sensing coverage requirement when the instant sensing and actuation tasks were undetected. The authors designed a two-stage task allocation algorithm that first clustered sensing locations and then allocated tasks to meet the coverage requirement [38].

B. UAV-assisted MCS

Compared with human participants, UAVs could move to desired places where human participants seldom or impossibly go. Furthermore, as they are equipped with various sensors, UAVs could complete more demanding tasks. Nowadays, UAVs have been exploited for numerous urban sensing applications, such as traffic conditions [39], monitoring air pollution [40], and parking lots [41]. However, UAVs have obvious constraints, such as a limited battery energy supply [42]. Authors in [43] re-utilized delivery UAVs in the crowdsensing scenario. In general, sensing tasks were assigned to UAVs according to the package delivery routes. Alkadi *et al.* argued that current UAV traffic management systems lacked a clear definition of secure interaction protocols. In order to solve the problem, they employed a mobile crowdsensing mechanism to enforce airspace rules and regulations [44]. Zhao *et al.* in [45] focused on solving how to motivate participants to perform tasks for the space-air-ground integrated vehicular crowdsensing systems. They proposed an incentive mechanism to eliminate the redundant number of selected participants. Wang *et al.* considered the joint use of trucks and UAVs for completing sensing tasks. In the process, the truck served as a mobile drone hub and task executor that swapped batteries for UAVs and collected sensing data [46]. Xie *et al.* proposed a UAV selection method that designed a UAV reputation incentive scheme and selected UAVs with high reputation value [47].

C. Deep reinforcement learn-based UAV trajectory scheme

Samir and Wu *et al.* were both proposed deep deterministic policy gradient (DDPG) methods for learning the trajectories of the deployed UAVs [17], [18]. Wang *et al.* proposed a trajectory planning method for multi-UAV scenario that employed deep Q-learning (DQN) to implement the method [24]. Fan *et al.* proposed a combined soft actor–critic (SAC) and alternating optimization (AO) algorithm to navigate a UAV to collect data [25]. Tao *et al.* aimed to leverage the package delivery activities of UAVs to solve the task allocation problem of MCS [26]. Authors employed a DDPG based trajectory planning method to delivery package. Dai, Wei and Oubbati *et al.* were proposed their own DRL based UAV trajectory schemes that employed UAVs to collect sensing data [19], [27], [28].

Compared with the aforementioned existing research, particularly the state-of-the-art methods discussed in Section II-C that utilize DRL for UAV navigation (as shown in Table I), our approach aims to enhance sensing data collection for meeting task sensing coverage requirements. We introduce a

hybrid MCS scenario wherein both human participants and UAVs collect sensing data simultaneously. Confronted with this intricate scenario, the task allocation method needs to distribute sensing tasks efficiently between human participants and UAVs. Here the proposed method considers human participants' tolerances and trajectories when allocating tasks to them, and seek to balance the trade-off among participants' tolerance level, limited budget, and the requirement for task sensing coverage. Additionally, to prevent the collection of redundant sensing data, the proposed method schedules UAV trajectories by taking into account the locations of unsensed tasks, human participants, obstacles, and other UAVs. We propose a multi-agent deep reinforcement learning framework for multi-UAV trajectory planning, which integrates the greedy method into DQN.

III. System Model

We consider a UAV-assisted MCS system that performs tasks such as air quality monitoring, temperature measurement, noise detection in a region. The system identifies several PoIs distributed across the region that require sensing. Subsequently, the system concurrently assigns tasks to both human participants and UAVs, as illustrated in Fig. 3. For human participants, the system predicts their trajectories using a reinforcement learning method, and allocates sensing tasks based on the Pareto optimal theory according their tolerance level and task coverage requirement. In the meantime, UAVs' trajectories are planned to perform tasks and avoid sensed PoIs, obstacles and other UAVs. In this context, a multi-agent deep reinforcement learning framework is utilized to plan trajectories, integrating the greedy method into deep Q-learning.

Consider there is a task associated with a certain budget B, and a number of PoIs P that distribute the sensing region needed to be sensed. The whole sensing campaign is divided into T time-slots with equal duration, as $\mathscr{T} = \{t|1, 2, \ldots, T\}$. 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. Thus each PoI needs to be sensed a maximum of Λ times by human participants or 1 time by UAVs in a single time-slot. λ_p^t indicates the total number of times a PoI p has been sensed up to time-slot t.

As human participant trajectory privacy represents a distinct category of personal privacy [48], measures are taken to minimize the potential leakage of personal information. Instead of disclosing the entirety of a human participant's trajectory, only the initial and destination locations are required. For a human participant i who wants to perform the task, he/she first claims his/her beginning and destination locations which are denoted by (l_i^b, l_i^d). Also, at the start of first time-slot, a human participant claims the reward they requested. We refer to $\mathscr{L}_i = \{x_{i,p}^t\}$ as the total set of PoIs that participant i has perceived, with $x_{i,p}^t = 1$ signifying that participant i has contributed sensing data to PoI p during time-slot t, otherwise $x_{i,p}^t = 0$. There is a tolerance threshold ψ_i for each human participant indicating the maximum number of pieces of sensing data that he/she could collect. And $r(\mathscr{L}_i)$ stands for the recruited participant i's final reward.

TABLE II
List of Important Notations

Notation	Explantation
B, P	Task budget, number of PoIs
T, $\mathscr{T}$	Number of time-slots, set of time-slots
Λ	Required sensing times of a PoI
λ_p^t	Total sensed times of PoI p up to time-slot t
l_i^b, l_i^d	Start and destination locations of human participant i
$\mathscr{L}_i$	Set of PoIs sensed by human participant i
$x_{\cdot,p}^t$	Parameter of whether a PoI p is sensed during time-slot t
ψ_i	Tolerance threshold of human participant i
$\mathscr{L}_\cdot$	Coverage set
$r(\mathscr{L}_i)$	Final reward of human participant i
θ_j^t, v_j^t	Direction and speed of UAV j
e_j^t	Power consumption of UAV j
$\mathscr{L}_j$	Set of PoIs sensed by UAV j

Following our former research [49], we consider sensing tasks to require all UAVs to fly around and cover PoIs. In the beginning, a UAV j is fully charged and its flight movements are influenced by two parameters $(\theta_j^t, v_j^t)_{j\in\mathscr{J}}$, θ_j^t is the direction and v_j^t is the speed. We consider the power consumption e_j^t of UAV j, which is directly related to the distance UAV j can fly, namely, $e_j^t = \gamma d_j^t$, and the battery capacity of a UAV j is E_j. The whole coverage set is $\mathscr{L}_j = \{x_{j,p}^t\}$, where $x_{j,p}^t = 1$ represents a UAV j that contributes a piece of high-quality sensing data at PoI p during time-slot t, and $x_{j,p}^t = 0$ otherwise.

The frequently used notations are summarized in Table II.

This paper aims to optimize the coverage completed ratio, which measures the amount of sensing data acquired to satisfy the coverage requirement while taking into account the budget and UAV energy limits.

$$
\begin{aligned}
\text{maximize: } & \frac{\sum_{t=1}^{T}\left|\bigcup_{i\in\{1,2,\ldots,I\}}\mathscr{L}_i\right|}{\Lambda * P * T} + \frac{\sum_{t=1}^{T}\left|\bigcup_{j\in\{1,2,\ldots,J\}}\mathscr{L}_j\right|}{P * T} \\
\text{subject to: } & \sum_{t=1}^{T}\sum_{i=1}^{I} r(\mathscr{L}_i) \le B, \\
& \mathscr{L}_i \le \psi_i, i \in \{1, 2, \ldots, I\}, t \in \{1, 2, \ldots, T\}, \\
& \sum_{t=1}^{T} e_j^t \le E_j,
\end{aligned}
\tag{1}
$$

where I and J is the number of recruiting participants and UAVs, respectively. $\sum_{t=1}^{T}\left|\bigcup_{i\in\{1,2,\ldots,I\}}\mathscr{L}_i\right|$ denotes the total contributed sensing data by human participants. $\sum_{t=1}^{T}\left|\bigcup_{i\in\{1,2,\ldots,I\}}\mathscr{L}_i\right|/\Lambda * P * T$ is the final coverage ratio sensed by human participants. The amount of sensing data collected by UAVs is denoted by $\sum_{t=1}^{T}\left|\bigcup_{j\in\{1,2,\ldots,J\}}\mathscr{L}_j\right|$.

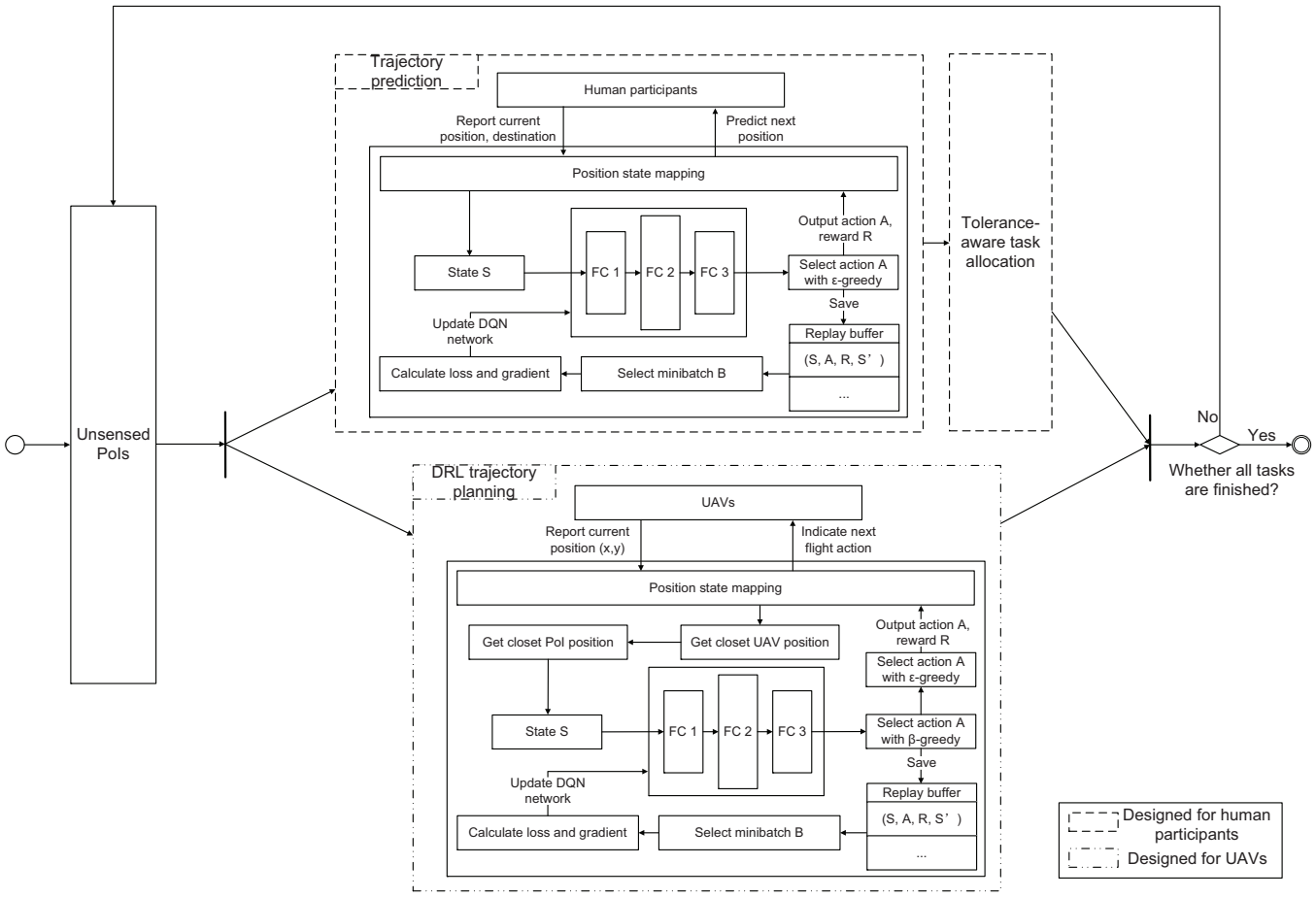

Fig. 3. Workflow of the proposed U-TAM.

Λ is the number of required times sensed from a PoI. $\sum_{t=1}^{T}\left|\bigcup_{j\in\{1,2,...,J\}}\mathscr{L}_j\right|/P*T$ is the final coverage ratio sensed by UAVs. ψ_i is the tolerance threshold for each human participant. The problem formulation of this paper is an NP-hard problem [49].

IV. Trajectory-based Tolerance-aware Task Allocation Method for Human Participants

As we mentioned in Section I, the method simultaneously allocates tasks to both UAVs and human participants according to their trajectories. In this section we describe the proposed method in order to maximize sensing coverage. We first focus on proposing the task allocation method for human participants. Taking into account the tolerance level of human participants, representing the maximum amount of sensing data they can contribute, along with the coverage requirement for each task, the task allocation method for the MCS system aims to maximize the sensed coverage while adhering to the tolerance constraints of human participants. This approach aligns with the Pareto optimal theory, asserting that the sensed coverage cannot be increased if a human participant shifts to sense another PoI, considering their tolerance level.

Specifically, the method first predicts the trajectories of human participants based on their start and destination locations. Then tasks are allocated to them following the Pareto optimal theory, which considers their tolerance level and the predicted trajectories. After that, we will combine human participant task allocation with UAV trajectory planning in the next section.

A. Human participant trajectory prediction method

As we mentioned in Section I, in order to allocate tasks efficiently, we predict the trajectory of human participants before allocating tasks to them. Similar to the digital navigation map, here we suppose that each human participant claims the start and end sites to the platform where he/she would like to contribute sensing data. Then the proposed method predicts the human participant's trajectory immediately and allocates task to him/her. Generally speaking, human participants are usually able to make an experienced judgment based on the actual congestion and choose roads with good road conditions. A route is more likely to be superior if there are more human participants who choose it. This indicates that the superiority of a route can usually be reflected in the number of times it is chosen. Therefore, better routes can be predicted based on a large number of reliable historical trajectories.

We formulate the trajectory prediction problem for human participants as a Markov Decision Process (MDP) and propose a reinforcement learning algorithm for predicting trajectories

based on historical GPS tracks, which is formulated as $T_r = \{< p_1, p_2, \ldots, p_n > | p_k = (lon_k, lat_k, tim_k)\}$, where lon_k, lat_k and tim_k are the longitude, latitude and time information, respectively. The road network is represented as a directed graph $G = (\mathscr{U}, \mathscr{S})$, where $\mathscr{U}$ is the full set of road junction nodes and $\mathscr{S}$ is the full set of all road sections in the road network. The junction node of an arbitrary road is denoted u_i, and s_{ij} denotes the road from node u_i to node u_j. We take the form of a limited latitude and longitude range, select certain intervals and applied road network matching to convert the sequence of GPS tracks of human participants into sequences of vertices in the graph where $T_r = \{< p_1, p_2, \ldots, p_n > | p_i = (u_i, tim_i)\}$.

Here the trajectory prediction problem is formulated as a MDP, which is defined as a tuple $(\mathscr{S}, \mathscr{A}, F, \mathscr{R})$, including: a set of states $\mathscr{S}$, a set of actions $\mathscr{A}$, the function of state transition probability $F : \mathscr{S} \times \mathscr{A} \times \mathscr{S} \to [0, 1]$ and the output reward function $\mathscr{S} \times \mathscr{A} \to \mathbb{R}$.

1) State space and observation space: $\mathscr{S} = \{u^k | u^k \in \mathscr{U}\}$ denotes the state set of an MDP, where u^k denotes the current road junction node u at hypothetical step k. Each human participant only knows his/her current junction node location u^k in graph G, which is called observation.

2) Action space: The action set is denoted by $\mathscr{A} = \{a^u | u \in \mathscr{U}\}$, where a^u is the next road junction that can be reached on the road junction node u.

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

4) Reward function: $\mathscr{S} \times \mathscr{A} \to \mathbb{R}$ expresses the expected immediate reward received after the state is transitioned from u^k to u^{k+1}.

Before introducing our reward function, we first denote a conditional turning probability under destination constraint by $P_r(u_j|d)$, which represents the probability that a human participant chooses u_j for the next junction node in order to reach the destination d. A simple way to calculate the value of $P_r(u_j|d)$ is to enumerate the road sections in all trajectories that end at d, then calculate the percentage of the trajectories that contain the road junction u_j. The expression is $P_r(u_j|d) = \frac{\sum T_r(u_j, d)}{\sum T_r(d)}$.

We normalize and employ $P_r(u^j|d)$ as a reward reference for making an action from u^k to u^{k+1}, specifically defined as follows,

$$r(s, a) = -(1 - P_r(u^{k+1}|d)) \times L(u^k, u^{k+1}),$$

where $L(u^k, u^{k+1})$ represents the length of the road from u^k to u^{k+1}. When the agent reaches the destination d, it will receive a larger reward based on the straight-line distance between its initial position and the destination d.

5)Problem formulation: Our problem can be formulated as:

$$Q^*(u^k, a^{u^k}) = Q(u^k, a^{u^k}) + \alpha\big(r + \gamma \max_{a^{u^k}} Q(u^{k+1}, a^{u^{k+1}}) - Q(u^k, a^{u^k})\big),$$

Algorithm 1 Deep Q-Networks in route prediction

Input: Start point u_0, end point u_e, max steps K
Output: Deep Q-Network $Q(s, a)$

1: Initialize replay memory D, its capacity is N;
2: Initialize action-value function Q with all zero weights;
3: **for** episode i in range $[1, episodes]$ **do**
4: Initialize $s_0 = u_0, k = 0$;
5: **while** current state $s_k! = u_e$ and step $k < K$ **do**
6: Select an action a_k with ϵ-greedy strategy that $a_k = arg\max_a Q^*(s_k, a_k)$;
7: Execute a_k, get reward r_k and new state s_{k+1};
8: Store (s_k, a_k, r_k, s_{k+1}) in D;
9: Sample random mini-batch of transitions (s_k, a_k, r_k, s_{k+1}) from D;
10: **if** $s_{k+1} = u_e$ **then**
11: $y_k = r_k$;
12: **else**
13: $y_k = r_k + \gamma \max_{a'}(s_{k+1}, a_{k+1})$;
14: **end if**
15: Train the Q-networks with $(y_k - Q(s_k, a_k))^2$ as the loss function;
16: $k = k + 1$;
17: **end while**
18: **end for**

and the optimal strategy of predicting a human participant's trajectory is given by $\pi = \arg\max_{a^u} Q^*(u^k, a^{u^k})$, where $\gamma \in (0, 1)$ represents the discount factor to show the importance between the future and present reward. This is a discrete Markov dynamic decision process. Here we use the DQN approach to predict the route chosen by the human participants. DQN combines the advantages of deep learning and reinforcement learning. In human road prediction, DQN can integrate different input data, such as historical travel probabilities and road distances, and learn through reinforcement learning how to make the most appropriate decisions in a given environment.

The prediction method is shown in Algorithm 1. The main processes are described as follows:

1) At the beginning, the algorithm initializes both the experience replay memory D with capacity N, and the action-value function Q with all zero weights (Line 1-2).
2) In Line 3-4, the agent which is initialized at position u_0 tries to find an empirical path to reach u_e for each episode. The step count of the agent is recorded by k.
3) Last, in Line 5-6, at the current episode the agent repeatedly chooses one of the neighboring rode junction nodes. Then the agent adds it to its route based on an ϵ-greedy strategy. Whenever the agent chooses such an action a_k, it receives an immediate reward r_k from the environment and gets the next state s_{k+1} (Line 7).
4) Finally, we store the values of (s_k, a_k, r_k, s_{k+1}) in the experience replay memory D, and take a number of samples from D to calculate the current target Q-Value y_k.

All parameters of the Q-Network are updated by means of gradient back propagation, which is based on the mean square

loss error function. After that, we extract a route as the most likely route, which is with the highest Q-value from the trained Q-Network.

B. Tolerance-aware task allocation method for human participants

If we ignore UAVs temporarily and only take human participants into account for the MCS campaigns, the target of this paper could be formulated as:

$$\begin{aligned} \text{maximize: } & \frac{\sum_{t=1}^{T}\left|\bigcup_{i\in\{1,2,\ldots,I\}}\mathscr{L}_i\right|}{\Lambda * P * T} \\ \text{subject to: } & \sum_{t=1}^{T}\sum_{i=1}^{I} r(\mathscr{L}_i) \le B, \\ & \mathscr{L}_i \le \psi_i, i \in \{1,2,\ldots,I\}, t \in \{1,2,\ldots,T\}. \end{aligned} \tag{2}$$

Here (2) can be transformed into a multi-objective optimization problem, which can be formulated as:

$$\begin{aligned} \text{maximize: } & f_j(x) = \begin{cases} 0, & x_j = 0 \\ 1, & x_j \neq 0 \end{cases} \\ \text{subject to: } & \sum_{i=1}^{I} r(x_i) \le B, \\ & \mathscr{L}_i \le \psi_i, i \in \{1,2,\ldots,I\}, \end{aligned}$$

where $x \in \{0,1,\ldots,I\}$ and $j \in \{1,2,\ldots,P\}$, I and P is the number of recruiting participants and PoIs, respectively. It is clear that PoI j is allocated to driver x_j when $f_j(x) = 1$. The budget for task is denoted by B. ψ_i is the tolerance threshold for each human participant.

For our multi-objective function optimal problems, assumption of a vector is $f(X) = \big(f_1(X), f_2(X), \ldots, f_n(X)\big)$. There are two decision variables X_u, X_v. Decision X_u is said to weak-dominate decision X_v if and only if $f(X_u) \le f(X_v)$ for all j in [0, J] and $f(X_u)$ is strictly better than $f(X_v)$ at least one objective. If there exists both i and j such that $f_i(X_u) < f_i(X_v)$ and $f_j(X_u) > f_j(X_v)$, thus X_u non-dominates X_v. The objective of our multi-objective optimization problem is to find a Pareto optimal solution, which defined as follows:

A solution X_u is a Pareto optimal solution if there is no other solution X_v, that dominates X_u.

Finding the Pareto optimal solution set means finding the Pareto optimal front. Here we employ a fast non-dominated sorting method to find the front. Briefly speaking, this method divides different PoIs allocation results into different sets, and repeats this step until the optimal set of non-dominated solutions is found [50]. The fast non-dominated sorting method is shown in Algorithm 2, which the time complexity is $\mathcal{O}(m \cdot N^2)$. We describe the main processes as follows:

1) The size of a task allocation set $\mathscr{P}$ is denoted by N. There are two parameters n_p and $\mathcal{S}_p$ needed to be calculated for each task allocation result p in the set. Where n_p represents the number of task allocation results for which Pareto dominates p, and $\mathcal{S}_p$ represents the set of task allocation results dominated by p. In the beginning, for each task allocation result $p \in \mathscr{P}$, initialize $n_p = 0$, and $\mathcal{S}_p = \emptyset$ (Line 3).
2) In Line 2-14, the method iterates through the entire task allocation set $\mathscr{P}$, calculates n_p and $\mathcal{S}_p$ for each task allocation result p, and places p with $n_p = 0$ into current non-dominated set $\mathscr{F}_1$.
3) The method iterates through $\mathscr{F}_i$ in Line 17-29. For each task allocation result p in $\mathscr{F}_i$, the value of n_q keeps to reduce as long as a task allocation result q exists in $\mathcal{S}_p$. If n_q is 0, then q will be added into a new non-dominated set.
4) The method continues to repeat step 3) until $\mathscr{F}_i == \emptyset$, where $\mathcal{S}_p = 0$.

Algorithm 2 Fast non-dominated sorting

Input: A task allocation set $\mathscr{P}$
Output: Different non-dominated sets stored by $\mathscr{F}$

1: $\mathscr{F}_1 = \emptyset$;
2: **for** $p \in \mathscr{P}$ **do**
3: $n_p = 0$, $\mathcal{S}_p = \emptyset$;
4: **for** $q \in \mathscr{P}$ **do**
5: **if** p Pareto dominates q **then**
6: $\mathcal{S}_p = \mathcal{S}_p \bigcup \{q\}$;
7: **else**
8: $n_p = n_p + 1$;
9: **end if**
10: **if** $n_p = 0$ **then**
11: $\mathscr{F}_1 = \mathscr{F}_1 \bigcup \{p\}$;
12: **end if**
13: **end for**
14: **end for**
15: $\mathscr{F} = \{\mathscr{F}_1\}$;
16: $i = 1$;
17: **while** $\mathscr{F}_i \neq \emptyset$ **do**
18: $\mathscr{F}_{i+1} = \emptyset$;
19: **for** $p \in \mathscr{F}_i$ **do**
20: **for** $q \in \mathcal{S}_p$ **do**
21: $n_q = n_q - 1$;
22: **if** $n_q = 0$ **then**
23: $\mathscr{F}_{i+1} = \mathscr{F}_{i+1} \bigcup \{q\}$;
24: **end if**
25: **end for**
26: **end for**
27: $\mathscr{F} = \mathscr{F} \bigcup \mathscr{F}_{i+1}$;
28: $i = i + 1$;
29: **end while**

The UAVs collaboration method is described in Algorithm 3. Here we suppose that $\mathscr{P}_t$ is the last task allocation set of the t_{th} generation, and $\mathscr{Q}_t$ is the offspring task allocation set that is generated by $\mathscr{P}_t$. They are of the same size N. We firstly combine $\mathscr{P}_t$ with $\mathscr{Q}_t$, then divide them into different sets $(\mathscr{F}_1, \mathscr{F}_2, \ldots, \mathscr{F}_n)$ using the non-dominated sorting algorithm described in Algorithm 2. The selection process starts at layer $\mathscr{F}_1$, and ends until its size attains or exceeds N for the first time. The main procedure is described as follows:

1) At the beginning, the algorithm initializes the final

Algorithm 3 Selection process of next generation $\mathscr{S}_t$

Input: Last task allocation set $\mathscr{P}_t$
Output: Next task allocation set $\mathscr{S}_t$
1: $\mathscr{S}_t = \emptyset$, $i = 1$;
2: $\mathscr{Q}_t$ = Recombination-Mutation($\mathscr{P}_t$);
3: $\mathscr{F}$ = Non-dominated-sorting($\mathscr{P}_t \bigcup \mathscr{Q}_t$);
4: **while** $|\mathscr{S}_t| + |\mathscr{F}_i| <= N$ **do**
5: **for** $p \in \mathscr{F}_i$ **do**
6: $\mathscr{S}_t = \mathscr{S}_t \bigcup p$;
7: **end for**
8: $i = i + 1$;
9: **end while**
10: **if** $|\mathscr{S}_t| < N$ **then**
11: The number of points that need to be picked out of $\mathscr{F}_i$: $K = N - |\mathscr{S}_t|$;
12: Select any K remaining allocation results on the NSGA-III reference point model;
13: **end if**

selecting set $\mathscr{S}_t = \emptyset$. Then the algorithm generates $\mathscr{Q}_t$ from $\mathscr{P}_t$ based on genetic recombination and polynomial variation (Line 1-2).

2) The result of $\mathscr{P}_t \bigcup \mathscr{Q}_t$ is separated into different sets by using the non-dominated-sorting method, which are stored by $\mathscr{F}$ (Line 3).
3) In Line 4-8, the algorithm adds all allocation results repeatedly from $\mathscr{F}_i$ to $\mathscr{S}_t$ until $|\mathscr{S}_t| + |\mathscr{F}_i| > N$.
4) The proposed algorithm selects N elements from $\mathscr{F}_i$ (Line 10-13).

After many generations of evolution, we finally choose the solution with the lowest budget cost from the N allocation results to be our optimal solution. The solution is the highest Pareto non-dominance, namely the Pareto optimal solution.

V. A TRAJECTORY SCHEDULING APPROACH FOR UAVS

In this section, we introduce a UAV trajectory scheduling approach that directs the UAVs to contribute sensing data jointly with human participants. The approach instructs the UAVs to sense data from PoIs which are rarely accessed by human participants, and all of UAVs to cooperatively collect PoIs while avoiding obstacles and other UAVs. It is worth noting that the trajectory scheduling method is 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.

Similar to the trajectory prediction method for human participants, the UAVs also need to choose their next actions based on the information of currently observing about their environment. We employ a reinforcement learning method to solve the UAV trajectory schedule problem. However, we also consider that as the number of UAVs are much fewer than that of PoIs in the UAV-assisted MCS context, the size of UAVs' observations can be effectively reduced by using the greedy method. Furthermore, the greedy method could also help UAVs avoid a lot of repetitive and inefficient exploration, and find the nearest PoIs quickly. Here we combine the deep Q-networks (DQN) with the greedy method to make trajectory schedules for UAVs.

1) The observation space: The observation space $\mathscr{S}$ of a UAV contains the following three parts, namely the UAV's own position, the relative position of the nearest PoI, and the relative position of the nearest UAV.

2) Action space: The action space $\mathscr{A}$ contains the actual flight actions that a UAV can perform, i.e., flying a number of distances in a certain direction (up, down, left, right, upper left, lower left, upper right, lower right).

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

4) Reward function: $\mathscr{S} \times \mathscr{A} \rightarrow \mathbb{R}$ expresses the expected immediate reward received after the state is transitioned from s^k to s^{k+1}. We define the reward received by the UAV u for making action a in state s as follows:

$$r^u(s, a) = c^u(s, a) - p^u(s, a),$$

where $c^u(s, a)$ denotes the reward that a UAV u can get by collecting PoIs after executing action a in the current observation s, the magnitude of the reward increases with the number of PoIs collected. $c^u(s, a)$ is define as:

$$c^u(s, a) = n \times v(p),$$

where n represents the number of PoIs collected by the UAV during its action a, while $v(p)$ denotes the value of a single PoI. If there are no PoIs to collect, the reward will be expressed as a negative relative distance from the UAV to the current nearest PoI. $p^u(s, a)$ denotes the penalty brought by the UAV doing action a in state s. If the UAV hits an obstacle or flies out of the boundary after executing the action a, $p^u(s, a)$ will incur a significant penalty value. Otherwise $p^u(s, a)$ will be calculated based on the distance between the UAV and its nearest uncollected PoI.

5) Problem Formulation: After the state transition function $\mathscr{F}$ and reward r are given by (3) and (4), our problem can be formulated as:

$$Q^{u,*}(s^k, a^k) = \mathop{\mathbb{E}}_{\substack{s^{k+1}\sim\mathscr{F} \\ \forall u\in\mathscr{U}}} \left[r^u(s^k, a^k) + \gamma \max_{a^{k+1}} Q^u(s^{k+1}, a^{k+1}) \right],$$

and our optimal strategy of UAVs is described as:

$$\pi^{u,*} = \mathop{\arg\max}_{\substack{a^k \\ \forall u\in\mathscr{U}}} \left[\mathop{\mathbb{E}}_{s^{k+1}\sim\mathscr{F}} [r^u(s^k, a^k) + \gamma \max_{a^{k+1}} Q^u(s^{k+1}, a^{k+1})] \right],$$

where $\gamma \in (0, 1)$ represents the discount factor to show the importance between the future and present reward. The UAV trajectory scheduling method is shown in Algorithm 4, which the main steps are expressed as follows.

1) At the beginning, the algorithm initializes both the experience replay memory buffer D with its capacity N, and the action-value function Q with all zero weights (Line 1-2).

Algorithm 4 UAV's trajectory scheduling method

Input: Start state s_{start}, Start power p_f, max steps K
Output: Deep Q-Network $Q(s, a)$

1: Initialize replay memory D, its capacity is N;
2: Initialize action-value function Q with all zero weights;
3: **for** episode i in range $[1, episodes]$ **do**
4: Initialize $s_0 = s_{start}, p = p_f, k = 0$;
5: **while** current power $p > 0$ and step $k < K$ **do**
6: Generate a random probability ϵ
7: **if** $\epsilon < \epsilon_0$ **then**
8: Generate a random probability $\beta \in [0, 1]$;
9: **if** $\beta < \beta_0$ **then**
10: Select an action a_g using a greedy method;
11: $a_k = a_g$;
12: **else**
13: Select an action a_r randomly;
14: $a_k = a_r$;
15: **end if**
16: **else**
17: Select an action a_k that $a_k = \max_a Q^*(s_k, a_k)$;
18: **end if**
19: Execute a_k, get reward r_k and new state s_{k+1};
20: Store (s_k, a_k, r_k, s_{k+1}) in D;
21: Sample random mini-batch of transitions (s_k, a_k, r_k, s_{k+1}) from D;
22: **if** $p \leq p_e$ or $k = K$ **then**
23: $y_k = r_k$;
24: **else**
25: $y_k = r_k + \gamma \max_{a'}(s_{k+1}, a_{k+1})$;
26: **end if**
27: Train the Q-networks with $(y_k - Q(s_k, a_k))^2$ as the loss function;
28: $k = k + 1$;
29: **end while**
30: **end for**

2) The UAV's initial state is denoted by s_0 and explores the environment based on the probability of ϵ. If the UAV decides to perform an exploratory action, it either chooses a greedy action with a probability of β, or performs behavior with the highest value of $Q^*(s_k, a_k)$ in the current network (Line 4-18).
3) The UAVs repeatedly choose the actions. After each action is performed, the feedback that is given by the environment will be stored in D (Line 19-20).
4) Finally, we choose a batch of samples from the experience replay memory D and update the Deep Q-Network based on the mean squared error loss function mentioned in the Algorithm 4 (Line 21-27).

VI. Performance Evaluation

This section begins by presenting the precise experimental setup along with the necessary parameters. The results are then discussed and compared to the widely used baselines.

TABLE III
Parameter of Settings

Parameters	Value
No. of human participants	Range from 20% to 100% of the total number of 200 participants, the default setting is 200
No. of UAVs	Range from 1 to 5, the default setting is 4
Max. flight distance of UAV	18 000m under an idealized condition
Speed of UAV	Range from 0m/s to 20m/s
Sensing range	Range from 12m to 20m, the default setting is 20m
No. of PoIs	Range from 170 to 270, the default setting is 270
The amount of budget	Range from 400 to 2 000 units, the default setting is 2 000 units
Amount of request reward	Range from 10 to 30 units Randomly
Tolerance level	Range from 3 to 5 pieces of sensing data Randomly
No. of requested data of each PoI	1 for UAVs and 5 for human participants
No. of time-slots	20

A. Setup

We employ a dataset of taxi movement trajectories as a reference for historical trajectories of human participants, which was collected in Rome, Italy [51]. The dataset contains 320 taxi drivers' travel tracks for one month, each track contains a number of records that include the taxi driver's ID, timestamp and GPS location (latitude and longitude). The parameter settings for our experiments are recorded in the Table III, some of these parameters are set following the configurations from work [49], including the number of UAVs and the quantity of requested data. We follow the work [18], [52] to set the parameters of signal transmission power, noise power spectral density and channel bandwidth. The subsequent procedure outlines the construction of our simulation platform.

- The dataset we employed for the simulation contains different areas of the city of Rome, of which we take $1\,000 \times 1\,000\text{m}^2$ as our simulation area. We scatter PoIs for every 50m on the area.
- We conduct the simulation experiments based on the DJI Mavic 2[1] drone, which the maximum sensing range is 20m and maximum flight distance is 18 000m under an idealized condition. To ensure safety, the speed of the drone is set between 0m/s and 20m/s.
- In the context of real-world environments with geographical constraints, UAV path planning becomes a highly complex task. UAVs are subject to various influences such as terrain occlusion, battery limitations, and channel noise. Taking into account the impact of channel noise during UAV flight, this paper, based on the settings of work [18], [52], proposes that each time the UAV performs a flight action or senses data, its battery consumption includes not only the energy required for the

[1]https://www.dji.com/be/mavic-2/info#specs

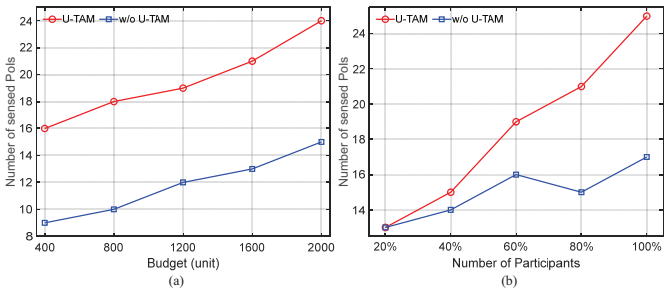

Fig. 4. Experiment on PoI sensed by tolerance-aware human participants, impact of (a) budget and (b) the number of human participants on the number of collected PoIs.

flight action but also an additional attenuation related to the transmission power (with a magnitude of 20dBm), a noise power spectral density of 10^{-17}W/Hz, and a channel bandwidth of 1MHz.

- The experiments are run on an Ubuntu 18.04.3 X64 server with a four-core, 3.60 GHz Intel(R) Xeon(R) Gold 5122 processor, 62GB of memory, and four Nvidia GeForce RTX 2080Ti graphics cards. Python 3.7 and Pytorch 1.7.0 are used to put the proposed method into practice. To train and evaluate the algorithms, we have created a unique reinforcement learning simulation environment that mimics OpenAI Gym, a traditional reinforcement learning environment.
- There are 270 PoIs prepared to be collected. For the UAVs, each PoI only needs to be collected one single time, whereas for the human participants, 5 times are required. The UAVs and human participants update their behavior synchronously once per time-slot.
- We set the requested reward and tolerance level of a human participant randomly in the range of $[10, 30]$ units and $[3, 5]$ pieces of sensing data, respectively.

B. Simulation Results

Firstly, for human participants, Fig. 4 is presented to evaluate whether the proposed method is tolerance-aware and takes sensing coverage into consideration. As the amount of sensing data should not exceed the tolerance level of human participant, how to allocate task to them becomes important. Fig. 4 illustrates that the proposed method collects more PoIs compared to the method that does not consider sensing coverage (referred to as "w/o U-TAM"). It is important to note that, in order to ensure data quality, a PoI is included in the count of successfully sensed PoIs when it is detected multiple times by human participants.

Next we show moving trajectories for $2, 3, 4, 5$ UAVs in Fig. 5. All of the UAVs successfully avoid obstacles and stay in the sensing areas. Because of the purposed U-TAM, each UAV has a clear direction to go and sense instead of flying around some PoIs and re-sensing them several times. Furthermore, with the increase number of UAVs, each one learns a new trajectory to sense, which clearly shows that UAVs could learn to collaborate with others. Because there were enough UAVs deployed and they were trained to work together, for

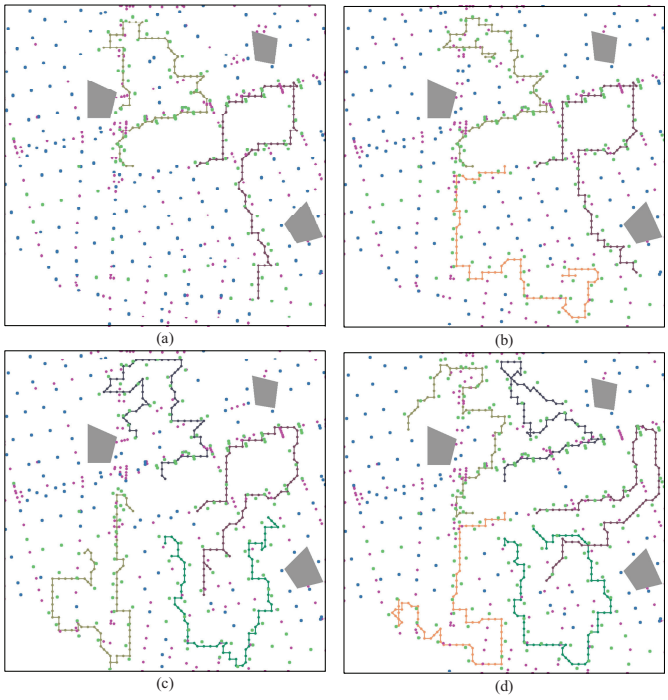

Fig. 5. UAV trajectories (lines for UAVs trajectories, gray blocks for obstacles, blue dots for un-sensed PoIs, and purple dots for sensed by participants).

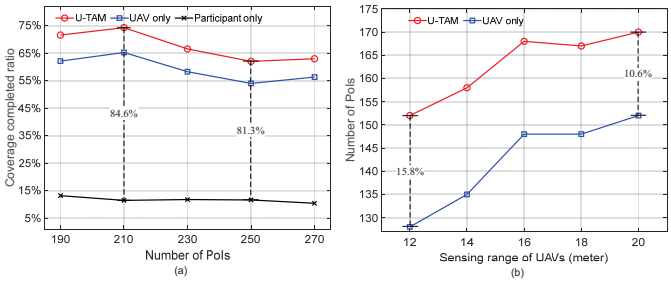

Fig. 6. (a) impact of number of PoIs on coverage completed ratio, and (b) impact of sensing range of UAVs on the number of PoIs.

instance, we can see from Fig. 3(d) that each UAV assumed responsibility for sensing a limited area and refrained from exploring beyond the range of other UAVs.

We conducted a comparative analysis of our human participant trajectory prediction method with two alternative approaches. The first method utilizes a fuzzy logic system to predict human participant trajectories [53]. This approach takes historical trajectory information of human participants as input and operates in three steps: converting crisp values into degrees of matching with linguistic values through membership functions, inferring fuzzy output based on predefined rules, and finally converting the fuzzy output into a crisp value using a typical center of gravity strategy. The second compared method is Dijkstra, a well-known route planning algorithm. Through experiments conducted on the same dataset, the accuracy results are 72.1% for our proposed method, 61.7% for the fuzzy logic system [53], and 48.7% for Dijkstra, respectively.

To compare with our proposed algorithm, we first employed

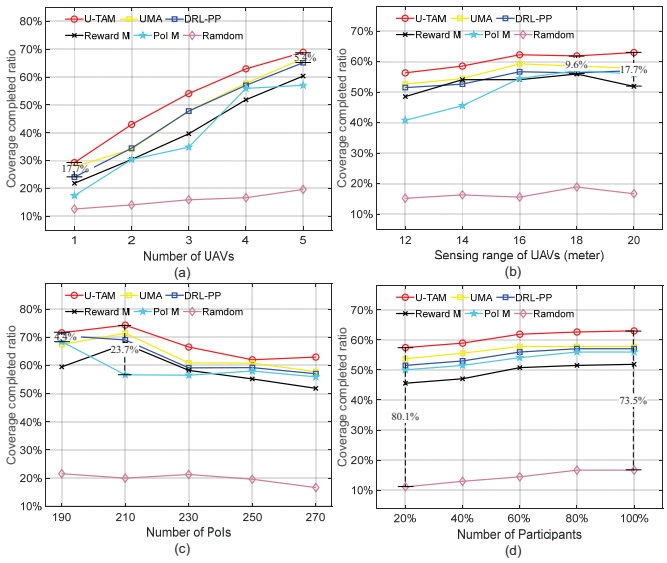

Fig. 7. Impact of (a) the number of UAVs, (b) sensing range of UAVs, (c) the number of PoIs, and (d) the number of participants on coverage completed ratio.

a single sensing medium to contribute data, i.e., UAVs or participants, which are referred to as "UAV only" and "Participant only", respectively. The impact of the number of PoIs on coverage completed ratio is shown in Fig. 6(a), where there are 4 UAVs sensing tasks. The proposed method always performs better than others. For example, U-TAM gains 81.3% and 84.6% more than that of participant only when the number of PoIs are 210 and 250, respectively. Fig.6(b) shows the impact of the sensing range of UAVs on the number of PoIs, where the number of PoIs increases with the sensing range extending. Here U-TAM improves 15.8% and 10.6% more PoIs than that of UAV only.

Next, we compared the proposed method with five baselines. The first one is UMA [49], which employs a reinforcement learning method named MADDPG to navigate UAVs (referred to as "UMA"). The second one is DRL-PP [27], which is likewise a reinforcement learning-based trajectory scheduling solution for UAVs (referred to as "DRL-PP"). A greedy approach is proposed as the third method that directs a UAV to sense a PoI in order to maximize the immediate reward (referred to as "Reward M"). The last method takes an action that maximizes the number of sensed PoIs (referred to as "PoI M"). The final one allows UAVs to take action randomly (referred to as "Random").

We can infer the following conclusions from Fig. 7: in terms of coverage completed ratio, U-TAM regularly performs better than all baselines. For instance, in Fig. 7(a), we can observe that the number of UAVs has a monotonic effect on the coverage completed ratio for all approaches. This is because having more UAVs enables better data collection. We observe that U-TAM gives 5.4% and 17.7% more than that of DRL-PP, when the number of UAVs is 5 and 1, respectively. The U-TAM similarly exhibits the highest performance in Fig. 7(b), giving 9.6% and 17.7% more than Reward M. In Fig. 7(c), when there are 210 and 190 PoIs, respectively,

TABLE IV
RESOURCE COST OF CPU AND MEMORY FOR PERFORMING U-TAM

Number of UAVs	2	3	4	5
Cost of CPUs (%)	100	99.7	100	100
Unified cost of CPUs (%)	25.0	24.9	25.0	25.0
Cost of memory (%)	3.6	3.6	3.6	3.7

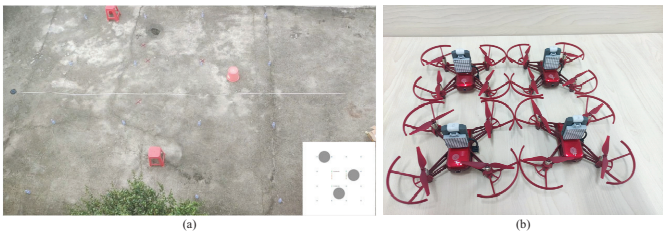

Fig. 8. (a) Experiment scenario, which area is $8 \times 8\text{m}^2$. We employ bottles and chairs to act as PoIs and obstacles, respectively. (b) 4 UAVs are employed to perform the real experiment, which model is DJI RoboMaster TT.

U-TAM improves the coverage completed ratio by 23.7% and 4.4% in comparison to PoI M. Finally, when there are 100% and 20% of human participants, respectively, U-TAM improves 73.5% and 80.1% when compared to Random, as shown in Fig. 7(d).

We finally measured how many resources were used to run the proposed algorithm when 2, 3, 4, or 5 UAVs were executing tasks. As shown in Tab. IV, our CPU utilization ratio is between 99.7% and 100%, while the maximum utilization ratio of our 4-core CPU is 400%. Here we unify the ratio of CPU into a range of $[0, 100\%]$ which is shown in the third line of Tab. IV. The memory utilization ratio remains at around 3.7%.

C. A small-scale real experiment for the proposed UAV trajectory scheduling approach

To further test the performance of the proposed method, a small scale real experiment was carried out on a $8 \times 8\text{m}^2$ area which is shown in Fig. 8(a). Here we employed 16 mineral water bottles as PoIs, and 3 plastic chairs and buckets to act as obstacles. We used 4 UAVs to perform tasks (shown in Fig. 8(b)). The main procedure of the experiment was that, the proposed method first calculated real time trajectories for each UAV, which were shown in the lower right screen. Then the UAVs flew correspondingly following the trajectory commands that were transmitted by Wi-Fi signals. All of the PoIs were sensed successfully at the end of the experiment. And none of the UAVs hit the obstacles or other UAVs.

The video can be downloaded at Baidu Netdisk [2] with the extraction code "utam".

VII. CONCLUSION AND FUTURE WORK

In order to maximize the sensing coverage, we proposed a UAV-assisted task allocation method (U-TAM) that allocated tasks to human participants and UAVs simultaneously in this paper. The UAVs coordinate with each other and improve the

[2]https://pan.baidu.com/s/1uVxYLSzUqcH20R-yyWtW9A

sensing coverage by complementing the sensing data provided by the human participants. In addition, the method took human participants' tolerance into consideration. With the purpose of finding the balance between task allocation requirements and human participants' tolerance level, the proposed U-TAM first predicted the trajectories of human participants and then allocated tasks to them following the Pareto optimal theory. The experiment results well justified the efficiency and robustness of the U-TAM in terms of coverage completed ratio, compared with the state-of-the-art.

The proposed hybrid sensing MCS system could find applications in smart city scenarios, where data can be easily sensed by human participants and UAVs to enhance the quality of life for citizens and improve public services in urban environments. The DRL based method facilitates UAV navigation for data sensing without human intervention. However, achieving the outstanding performance of DRL comes with challenges, requiring not only a substantial amount of training data but also a large number of network parameters and significant computational power. It becomes crucial to explore alternative strategies for running the DRL-based method locally when network connections to servers fail. Mobile devices emerge as potential solutions, given their ubiquity in daily life. Nevertheless, most mobile devices have limitations in computational resources. In the future, we anticipate addressing the challenge of deploying DRL models on mobile devices. This could involve leveraging model compression technologies to reduce computational resource requirements, storage overhead, and inference time.

REFERENCES

[1] T. Zhou, Z. Cai, K. Wu, Y. Chen, and M. Xu, "Fidc: A framework for improving data credibility in mobile crowdsensing," *Computer Networks*, vol. 120, pp. 157–169, 2017.

[2] 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.

[3] W. Liu, L. Wang, E. Wang, Y. Yang, D. Zeghlache, and D. Zhang, "Reinforcement learning-based cell selection in sparse mobile crowdsensing," *Computer Networks*, vol. 161, pp. 102–114, 2019.

[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] Y. Liu, Z. Yu, H. Cui, S. Helal, and B. Guo, "Safecity: A heterogeneous mobile crowd sensing system for urban public safety," *IEEE Internet of Things Journal*, 2023.

[6] 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.

[7] 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.

[8] J. Wang, J. Liu, and G. Zhao, "Two-phased participant selection method based on partial transfer learning in mobile crowdsensing," *ACM Transactions on Sensor Networks*, vol. 19, no. 2, pp. 1–17, 2022.

[9] 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.

[10] 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.

[11] 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.

[12] J. Wang, Y. Wang, D. Zhang, and S. Helal, "Energy saving techniques in mobile crowd sensing: Current state and future opportunities," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 164–169, 2018.

[13] C. Xu and W. Song, "Intelligent task allocation for mobile crowdsensing with graph attention network and deep reinforcement learning," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 2, pp. 1032–1048, 2023.

[14] Y. Huang, H. Chen, G. Ma, K. Lin, Z. Ni, N. Yan, and Z. Wang, "Opat: Optimized allocation of time-dependent tasks for mobile crowdsensing," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2476–2485, 2021.

[15] H. Wang, C. H. Liu, H. Yang, G. Wang, and K. K. Leung, "Ensuring threshold aoi for uav-assisted mobile crowdsensing by multi-agent deep reinforcement learning with transformer," *IEEE/ACM Transactions on Networking*, pp. 1–16, 2023.

[16] W. Jiang, B. Ai, C. Shen, M. Li, and X. Shen, "Age-of-information minimization for uav-based multi-view sensing and communication," *IEEE Transactions on Vehicular Technology*, pp. 1–15, 2023.

[17] M. Samir, C. Assi, S. Sharafeddine, D. Ebrahimi, and A. Ghrayeb, "Age of information aware trajectory planning of uavs in intelligent transportation systems: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 382–12 395, 2020.

[18] S. Wu, W. Xu, F. Wang, G. Li, and M. Pan, "Distributed federated deep reinforcement learning based trajectory optimization for air-ground cooperative emergency networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 8, pp. 9107–9112, 2022.

[19] Z. Dai, C. H. Liu, R. Han, G. Wang, K. Leung, and J. Tang, "Delay-sensitive energy-efficient uav crowdsensing by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021, doi: 10.1109/TMC.2021.3113052.

[20] B. Alzahrani, O. S. Oubbati, A. Barnawi, M. Atiquzzaman, and D. Alghazzawi, "Uav assistance paradigm: State-of-the-art in applications and challenges," *Journal of Network and Computer Applications*, vol. 166, p. 102706, 2020.

[21] K. Messaoudi, O. S. Oubbati, A. Rachedi, A. Lakas, T. Bendouma, and N. Chaib, "A survey of uav-based data collection: Challenges, solutions and future perspectives," *Journal of Network and Computer Applications*, vol. 216, p. 103670, 2023.

[22] J. Rubio-Aparicio and J. Santa, "An embedded crowdsensing unit for mobile urban pollution monitoring," *IEEE Communications Magazine*, vol. 61, no. 1, pp. 90–96, 2022.

[23] T. A. N. Dinh, A. D. Nguyen, T. T. Nguyen, T. H. Nguyen, and P. Le Nguyen, "Spatial-temporal coverage maximization in vehicle-based mobile crowdsensing for air quality monitoring," in *Proc. IEEE WCNC 2022*. IEEE, 2022, pp. 1449–1454.

[24] X. Wang, M. C. Gursoy, T. Erpek, and Y. E. Sagduyu, "Learning-based uav path planning for data collection with integrated collision avoidance," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16 663–16 676, 2022.

[25] X. Fan, M. Liu, Y. Chen, S. Sun, Z. Li, and X. Guo, "Ris-assisted uav for fresh data collection in 3d urban environments: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 632–647, 2022.

[26] X. Tao and A. S. Hafid, "Trajectory design in uav-aided mobile crowdsensing: A deep reinforcement learning approach," in *Proc. ICC 2021*, 2021, pp. 1–6.

[27] K. Wei, K. Huang, Y. Wu, Z. Li, H. He, J. Zhang, J. Chen, and S. Guo, "High-performance uav crowdsensing: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, 2022.

[28] O. S. Oubbati, M. Atiquzzaman, H. Lim, A. Rachedi, and A. Lakas, "Synchronizing uav teams for timely data collection and energy transfer by deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 6682–6697, 2022.

[29] H. Zhu, T. Shou, R. Guo, Z. Jiang, Z. Wang, Z. Wang, Z. Yu, W. Zhang, C. Wang, and L. Chen, "Redpacketbike: A graph-based demand modeling and crowd-driven station rebalancing framework for bike sharing systems," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022, doi: 10.1109/TMC.2022.3145979.

[30] B. Zhao, X. Liu, W.-N. Chen, and R. Deng, "Crowdfl: Privacy-preserving mobile crowdsensing system via federated learning," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022, doi: 10.1109/TMC.2022.3157603.

[31] Y. Ding, L. Zhang, and L. Guo, "Dynamic delayed-decision task assignment under spatial-temporal constrains in mobile crowdsensing,"

IEEE Transactions on Network Science and Engineering, vol. 9, no. 4, pp. 2418–2431, 2022.

[32] J. Wang, L. Wang, Y. Wang, D. Zhang, and L. Kong, "Task allocation in mobile crowd sensing: State-of-the-art and future opportunities," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3747–3757, 2018.

[33] Y. Zhang, Z. Ying, and C. L. P. Chen, "Achieving privacy-preserving multi-task allocation for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 16 795–16 806, 2022.

[34] M. Xiao, W. Jin, C. Li, and M. Li, "Eliciting joint truthful answers and profiles from strategic workers in mobile crowdsourcing systems," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022, doi: 10.1109/TMC.2022.3159228.

[35] J. Xu, Z. Luo, C. Guan, D. Yang, L. Liu, and Y. Zhang, "Hiring a team from social network: Incentive mechanism design for two-tiered social mobile crowdsourcing," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022, doi: 10.1109/TMC.2022.3162108.

[36] M. Huang, V. C. Leung, A. Liu, and N. N. Xiong, "Tma-dpso: Towards efficient multi-task allocation with time constraints for next generation multiple access," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1652–1666, 2022.

[37] L. Wang, Z. Yu, K. Wu, D. Yang, E. Wang, T. Wang, Y. Mei, and B. Guo, "Towards robust task assignment in mobile crowdsensing systems," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022, doi: 10.1109/TMC.2022.3151190.

[38] H. Yin, Z. Yu, L. Wang, J. Wang, L. Han, and B. Guo, "Isiatasker: Task allocation for instant-sensing-instant-actuation mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3158–3173, 2022.

[39] H. Huang, A. V. Savkin, and C. Huang, "Decentralized autonomous navigation of a uav network for road traffic monitoring," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2558–2564, 2021.

[40] Y. Liu, J. Nie, X. Li, S. H. Ahmed, W. Y. B. Lim, and C. Miao, "Federated learning in the sky: Aerial-ground air quality sensing framework with uav swarms," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9827–9837, 2021.

[41] P. Gogoi, J. Dutta, R. Matam, and M. Mukherjee, "An uav assisted multi-sensor based smart parking system," in *Proc. IEEE INFOCOM 2020 Workshops*, 2020, pp. 1225–1230.

[42] L. Ding, D. Zhao, M. Cao, and H. Ma, "When crowdsourcing meets unmanned vehicles: Toward cost-effective collaborative urban sensing via deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12 150–12 162, 2021.

[43] C. Xiang, Y. Zhou, H. Dai, Y. Qu, S. He, C. Chen, and P. Yang, "Reusing delivery drones for urban crowdsensing," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021, doi: 10.1109/TMC.2021.3127212.

[44] R. Alkadi and A. Shoufan, "Unmanned aerial vehicles traffic management solution using crowd-sensing and blockchain," *ArXiv*, vol. abs/2110.14979, 2021.

[45] R. Zhao, L. T. Yang, D. Liu, X. Deng, and Y. Mo, "A tensor-based truthful incentive mechanism for blockchain-enabled space-air-ground integrated vehicular crowdsensing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2853–2862, 2022.

[46] Z. Wang, B. Zhang, and C. Li, "Joint path planning of truck and drones for mobile crowdsensing: Model and algorithm," in *Proc. IEEE GLOBECOM 2021*, 2021, pp. 1–6.

[47] L. Xie, Z. Su, N. Chen, and Q. Xu, "Secure data sharing in uav-assisted crowdsensing: Integration of blockchain and reputation incentive," in *Proc. IEEE GLOBECOM 2021*, 2021, pp. 1–6.

[48] W. Wang, Y. Wang, P. Duan, T. Liu, X. Tong, and Z. Cai, "A triple real-time trajectory privacy protection mechanism based on edge computing and blockchain in mobile crowdsourcing," *IEEE Transactions on Mobile Computing*, vol. 22, no. 10, pp. 5625–5642, 2023.

[49] H. Gao, J. Feng, Y. Xiao, B. Zhang, and W. Wang, "A uav-assisted multi-task allocation method for mobile crowd sensing," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022, doi: 10.1109/TMC.2022.3147871.

[50] H. Li, K. Deb, Q. Zhang, P. N. Suganthan, and L. Chen, "Comparison between moea/d and nsga-iii on a set of novel many and multi-objective benchmark problems with challenging difficulties," *Swarm and Evolutionary Computation*, vol. 46, pp. 104–117, 2019.

[51] 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.

[52] L. Fu, Z. Zhao, G. Min, W. Miao, L. Zhao, and W. Huang, "Energy-efficient 3d data collection for multi-uav assisted mobile crowdsensing," *IEEE Transactions on Computers*, vol. 72, no. 7, pp. 2025–2038, 2023.

[53] J. Zhang and X. Zhang, "Multi-task allocation in mobile crowd sensing with mobility prediction," *IEEE Transactions on Mobile Computing*, vol. 22, no. 2, pp. 1081–1094, 2023.

Xinbin Liu received the B.S. degree from Beijing University of Posts and Telecommunications in 2021, where he is currently pursuing the master's degree with the School of Computer Science (National Pilot Software Engineering School). His research interests include mobile crowd sensing and Internet-of-Things.

Ye Wang received the B.S.degree from the Beijing University of Posts and Telecommunications in 2022, where he is currently pursuing the master's degree with the School of Computer Science (National Pilot Software Engineering School). His research interests include applied mobile crowdsensing, deep learning, and meta learning.

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.

Edith C.H. Ngai is currently an Associate Professor in the Department of Electrical and Electronic Engineering, The University of Hong Kong. Before joining HKU in 2020, she was an Associate Professor in the Department of Information Technology at Uppsala University, Sweden. Her research interests include Internet-of-Things, machine learning, data analytic, and smart cities. She received a Meta Policy Research Award in Asia Pacific in 2022. She was elected as one of the IEEE N^2Women Stars in Computer Networking and Communications in 2022. She is an IEEE ComSoc Distinguished Lecturer in 2023-2024.

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.

Chuhan Wang is currently pursuing the B.S.degree with School of Physics and Optoelectronic Engineering, Guangdong University of Technology. Her research interests include control and automation, and mobile crowdsensing.

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.