Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.059880

ARTICLE

MARCS: A Mobile Crowdsensing Framework Based on Data Shapley Value Enabled Multi-Agent Deep Reinforcement Learning

Yiqin Wang[1], Yufeng Wang[1,*], Jianhua Ma[2] and Qun Jin[3]

[1]School of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, 210046, China
[2]Faculty of Computer & Information Sciences, Hosei University, Tokyo, 1848584, Japan
[3]Faculty of Human Sciences, School of Human Sciences, Waseda University, Tokyo, 1698050, Japan

*Corresponding Author: Yufeng Wang. Email: wfwang@njupt.edu.cn

Received: 18 October 2024; Accepted: 18 December 2024; Published: 06 March 2025

ABSTRACT: Opportunistic mobile crowdsensing (MCS) non-intrusively exploits human mobility trajectories, and the participants' smart devices as sensors have become promising paradigms for various urban data acquisition tasks. However, in practice, opportunistic MCS has several challenges from both the perspectives of MCS participants and the data platform. On the one hand, participants face uncertainties in conducting MCS tasks, including their mobility and implicit interactions among participants, and participants' economic returns given by the MCS data platform are determined by not only their own actions but also other participants' strategic actions. On the other hand, the platform can only observe the participants' uploaded sensing data that depends on the unknown effort/action exerted by participants to the platform, while, for optimizing its overall objective, the platform needs to properly reward certain participants for incentivizing them to provide high-quality data. To address the challenge of balancing individual incentives and platform objectives in MCS, this paper proposes MARCS, an online sensing policy based on multi-agent deep reinforcement learning (MADRL) with centralized training and decentralized execution (CTDE). Specifically, the interactions between MCS participants and the data platform are modeled as a partially observable Markov game, where participants, acting as agents, use DRL-based policies to make decisions based on local observations, such as task trajectories and platform payments. To align individual and platform goals effectively, the platform leverages Shapley value to estimate the contribution of each participant's sensed data, using these estimates as immediate rewards to guide agent training. The experimental results on real mobility trajectory datasets indicate that the revenue of MARCS reaches almost 35%, 53%, and 100% higher than DDPG, Actor-Critic, and model predictive control (MPC) respectively on the participant side and similar results on the platform side, which show superior performance compared to baselines.

KEYWORDS: Mobile crowdsensing; online data acquisition; data Shapley value; multi-agent deep reinforcement learning; centralized training and decentralized execution (CTDE)

1 Introduction

With the explosive popularity of smart devices (such as smartphones and smart wearables, etc.) that are equipped with unprecedented sensing, computing, and communication capabilities, recently, Mobile Crowdsensing (MCS) has already become a promising data acquisition paradigm, especially for collecting urban sensing data, in which MCS participants conduct various microtasks of data sensing released by data platform on behalf of data requester. It allows the abundant number of MCS participants to share the local geospatial information and knowledge acquired by their sensor-enhanced devices. Different from statically

deployed sensing infrastructures, participants' mobility makes MCS a versatile platform and enables a broad range of applications, including environment monitoring urban management, etc. [1].

Generally, MCS can be categorized into two paradigms: opportunistic and participatory. In opportunistic MCS, participants do not need to specify the crowdsensing tasks to be completed. Compared with participatory MCS, Opportunistic MCS is a more attractive, user-friendly, and cost-effective sensing paradigm, since it does not affect crowdworkers' daily routines [2]. To characterize the sensing opportunity and quality, some metrics were proposed, including opportunistic coverage, etc. [3].

Basically, the MCS system is composed of different stakeholders with different or even conflicting interests: participants aim to earn more money through strategic choice, e.g., putting in less effort. MCS platform endeavors to acquire high-quality crowd-sensed data at a lower cost. Naturally, various game theory models have been used to characterize and align the interactions between participants and platforms, including cooperative games, Stackelberg games, auction theory, etc. [4]. An auction-based bi-objective robust mobile crowdsensing system is modeled, which aims to maximize both the expected benefit and coverage [5].

Considering the uncertainty of participants executing tasks in opportunistic MCS by assuming the specific probability model of users' coverage, the value maximization problem in opportunistic MCS was transformed into an ordered submodularity value function model with budget constraints [6]. Specifically, the properties of the ordered submodular model are combined with reverse auction models to solve the allocation and payment problems.

The works above are essentially pure model-based MCS framework, which mathematically characterizes participants' behaviors with extremely simplified but unrealistic models. The weak point is that they did not fully consider participants' behaviors under complex uncertainties and fast-changing environments. Practically, MCS participants face great uncertainties that stem from the sensing environment and the implicit and explicit interactions with other participants and the MCS data platform respectively. In particular, each participant cannot observe others' actions but has to decide her own actions in sensing tasks only based on locally observed information. In addition, the economic returns provided by the MCS platform to participants not only depend on their own efforts but also on the actions of other participants. Therefore, it is imperative to investigate how to make sequential decisions from the perspective of participants to optimize each participant's long-term return, when facing uncertainties.

Since Deep reinforcement learning (DRL) integrates the expressive representation ability of deep learning methods, with sequential decision-making ability under uncertainties of reinforcement learning (RL), recently, DRL has witnessed great application in various fields [7]. In [8], a single-agent RL algorithm is proposed to incentivize MCS participants. Among the approaches, Multi-Agent Reinforcement Learning (MARL) has witnessed great popularity, due to its excellent ability to learn without knowing the dynamic world model [9,10].

In this paper, a multi-agent DRL-based online geospatial MCS framework MARCS is proposed to incorporate the uncertainty of participants' decision-making in the real user's mobility and sensing environment. This framework takes into account the interests of the participant and the platform: the platform can seek to maximize the overall value of all crowdsourced data, and each participant can optimize her long-term return by making a sequential decision based on the locally observed information. Specifically, the main contributions of this paper are given as follows:

- The proposed participant-centric MCS paradigm treats each MCS participant as a learning agent and uses the multi-agent deep deterministic policy gradient (MADDPG) algorithm based on the Actor-Critic architecture. This enables each agent to learn how to adjust its sensing effort level to maximize long-term

returns, while also ensuring the provision of high-quality sensing data that aligns with the platform's objectives. To handle the non-stationarity and partial observability challenges, the centralized training and decentralized execution (CTDE) approach is employed, where the Critic uses global information during training to improve learning, and the Actor makes decisions based on local observations during execution. This ensures efficient adaptation in dynamic and partially observable environments.

- The proposed approach incorporates the platform's overall objective by considering both the active factors of participants' trajectories and the quality of the reported data. Active factors represent the contribution of each participant's trajectory to the platform's objectives, while the quality of reported data reflects the sensing effort strategically determined by each agent. To align individual efforts with system goals, the platform employs the Shapley value from cooperative game theory to efficiently estimate the value of data contributed by each agent, which serves as the immediate payment to the agent. Then, payment minus the sensing cost of each participant as the reward signal is used to guide the training of local agents. The Shapley value's advantage lies in its ability to capture the relationships among agents without requiring prior knowledge, thereby enhancing the effectiveness of learning and collaboration among participants.
- To verify that the MARCS scheme can achieve higher benefits on both sides of the participant and platform, we perform extensive simulations using real-world user mobility trajectory dataset, and compared to single agent reinforcement learning methods (AC and DDPG) and model predictive control (MPC) methods, demonstrating the feasibility and superiority of the proposed scheme.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 presents the proposed MARCS framework and its four modules respectively located on the participant and platform sides. Section 4 describes the participants' real mobility trajectory dataset, presents the detailed experiment settings, and comprehensively compares our proposal with the independent DRL agents-based schemes and MPC method. Finally, we briefly conclude this paper.

2 Related Works

One of the key issues in opportunistic MCS is to motivate users to participate in sensing tasks, as executing tasks may consume resources and incur costs for participants. Therefore, in literature, significant efforts have been made to design incentive mechanisms. Zhang et al. [11] considered the opportunistic nature of participants and proposed three online incentive mechanisms based on reverse auctions. A group-based sensing framework was proposed by [12], which allows sensing data to be stored and processed locally on user devices and then exchanged information between mobile users in a P2P mode. To provide necessary incentives for participants, a quality-aware data-sharing market is proposed, in which mobile users' behavior dynamics are analyzed from the game-theoretic perspective, and the existence and uniqueness of the game equilibrium are characterized. Similarly, Cao et al. [13] designed a game theory-based MCS incentive mechanism to incentivize nearby appropriate user devices to share sensing task resources, in which an auction-based task migration algorithm is proposed to adjust resources among mobile devices for a better crowdsensing response. Zhan et al. [14] proposed an incentive mechanism based on bargaining between the sensing platform and users, which is solved by a distributed iterative algorithm. In [15], the sensed data trading was formulated as a two-person cooperative game, which is solved by the Nash bargaining theory.

Guo et al. [16] proposed a multi-task MCS participant selection framework, ActiveCrowd, for tasks with time requirements or time delays. A multi-task allocation problem with time constraints is also investigated in [17] to maximize the utility of the MCS platform, in which two heuristically evolutionary algorithms are designed to solve this problem.

In summary, most of the above works adopt model-based optimization or heuristic schemes, which can't deal with complex MCS environments and participants' interactions.

With the development of learning technology, more and more research has applied reinforcement learning technology to MCS scenarios. For the scenario of vehicle crowdsensing, the interaction between the platform and the vehicles is modeled as a congestion sensing game, and tabular Q-learning algorithms are respectively used by the platform and vehicles to make corresponding decisions [18]. However, the tabular RL only works for small discrete action spaces. Moreover, in this work, each vehicle independently uses the Q-learning technique to derive its sensing strategy, which can't incorporate the implicit interactions among participants and may hamper the learning performance. Without assuming that the platform can model/predict the movement of participants before selecting participants for geospatial data sensing, our previous work [19] proposed a data-driven RL-based, i.e., an improved Q-learning-based online participant selection scheme to optimize the sensing coverage ratio and degree of geospatial area.

Recent deep reinforcement learning algorithms have been applied to incentive design and task allocation in MCS. In [20], without knowing the private information of the mobile users, a DRL-based dynamic incentive mechanism formulated with Stackelberg game theory was proposed to learn the optimal pricing strategy by MCS service providers directly from the game experience.

These works in literature were mainly platform-centric: that is the platform acts as the decision-making agent to optimize its own benefit through participant selection, task allocation, incentive mechanism, etc. However, they ignore that multiple participants are all decision-making agents who independently put effort and take action, especially in opportunistic MCS. Zhan et al. [21] characterize the interactions between multiple mobile users and multiple data requesters as MCS game and design a DRL-based dynamic incentive mechanism to enable the data requesters to learn the optimal pricing strategies directly from game experiences. Specifically, the MCS game is formulated as a multi-agent Markov decision processes (MDPs), in which each data requester acts as an agent and the environment consists of multiple mobile users, and the AC framework based on policy gradient method is independently used by each data requesters for decision-making. The work is essentially MCS platform centric, and the main disadvantage is that each agent only uses the locally observed information to learn the pricing strategy, which totally ignores the implicit relationships among agents, and may hamper the performance. Xu et al. [22] incorporate a graph attention network into DRL to train with different problem instances, which aims to explore the best solutions better for task allocation problems from a platform perspective. The work also focuses on maximizing the benefit of the platform, which does not consider the potential interactions between the participants. From the perspective of MCS participants, an intelligent crowdsensing algorithm IntelligentCrowd was developed, based on multi-agent deep reinforcement learning (MADRL) [23], in which the MCS participants' behaviors are modeled with multi-agent MDPs. However, this work didn't consider the platform's overall objective through data acquisition. It didn't explore the interactions between the platform's mechanism design and MCS participants' decision-making. Similarly, Dongare et al. [24] propose a federated DRL method to enable each agent to learn a task participation strategy, which focuses on the interactions between participants. The benefits of the platform side and the implicit relations between the platform and participants are less stressed. Xu et al. [25] propose a decentralized DRL method that can leverage redundant computation and network resources of workers' devices. Although a decentralized way can reduce the costs of the platform, it lacks a unique design and the consideration of benefit for the platform side.

A subclass of cooperative games in MADRL related to our work is the so-called global reward game, where all agents aim to maximize the global reward. However, simply assigning each agent the shared global reward can't correctly reflect each agent's contribution to the global group/coalition, which is also known credit assignment problem. Shapley value is one of the most popular methods to fairly distribute the team's

payoff to each agent by considering the extent to which the agent increases the marginal contributions of the coalitions in all possible permutations. A lot of work exists to leverage Shapley to address the issue. For example, a local reward approach called the Shapley Q-value was proposed to divide the global reward into local rewards according to each agent's own contribution, and the Shapley Q-value deep deterministic policy gradient method is designed as the Critic for each agent [26]. Inspired by the idea, our paper exploits Shapley value to assign global benefits to participants according to their contribution, to encourage high-quality data acquirement.

Compared to model-based methods, our model-free MADRL approach combined with Shapley value reduces the reliance on explicit environment modeling, allowing it to better adapt to the complexity and dynamics of MCS scenarios. Integrating Shapley value for reward allocation ensures fair compensation based on each agent's contribution, overcoming the limitations of heuristic designs and enhancing learning efficiency in multi-agent systems. Besides, different from the above work, our paper focuses on characterizing implicit interactions among multiple MCS participants, as well as explicit interactions between the data platform and MCS participants, and aims at incentivizing some appropriate participants to provide high-quality data, when MCS participants play non-cooperative game with only partially observed information.

In a sense, the interaction between the data platform and MCS participants is economically equivalent to the principal-(multiple) agent relationship in the economic field: The principal buys some good or service from these agents, which is the outcome of each agent's productive action/effort. However, the key issue is only the outcome not the effort can be observed, that is, the agent's action is hidden: each agent knows what action she has taken but the principal does not directly observe her action, so-called hidden action or moral hazard. In addition, participants' economic return given by the principal simultaneously depends both on their own efforts and other participants' decisions. To deal with this problem, based on the principal's overall objective, a reward-sharing rule should be designed, in which the principal can indirectly influence the agent's hidden actions by compensating the agent contingent on the consequences of her actions [27]. Originating from cooperative game theory, Shapley value is a widely used rule to distribute benefits reasonably and fairly by accounting for the corresponding contribution of all game players. Shapley value satisfies the properties of efficiency, symmetry, nullity, linearity, and coherency. Since the Shapley value of each player is the average marginal contribution of the player to the benefit of the grand coalition computed from all permutations, the relations among the players are considered sufficiently without prior knowledge, which further promotes the learning of local agents. Due to these axiomatic properties, Shapley value has found wide application in machine learning, including feature selection, data valuation, model explainability, multi-agent reinforcement learning, etc. [28].

3 MARCS: The Proposed Data Shapley Value Enabled DRL-Based MCS Scheme

Fig. 1 illustrates the workflow of our proposed data Shapley value-enabled DRL-based MCS framework MARCS in one sensing round, which includes the MCS participant side and data platform side. Specifically, the number marked in each module in Fig. 1 denotes the operating sequence of a sensing round, which is from ① to ④. Considering an MCS system with a set of mobile participants $U = \{u_1, u_2, \ldots, u_N\}$ and a data platform, there are two-way communications between mobile users and the platform. Each participant has the capability of sensing some data in certain grids at each sensing round/timestep.

According to the MCS requirement, the target region R is gridizated: divided into m common grids and n hotspot grids, represented as: $R = \{g_1, \ldots, g_i, \ldots, g_m\} \cup \{h_1, \ldots, h_j, \ldots, h_n\}$, where $g_i, i = \{1, \ldots, m\}$ denotes the common grid, and $h_j, j = \{1, \ldots, n\}$ is the hotspot grid. The MCS platform intends to collect more sensing data about hotspot grids than common grids.

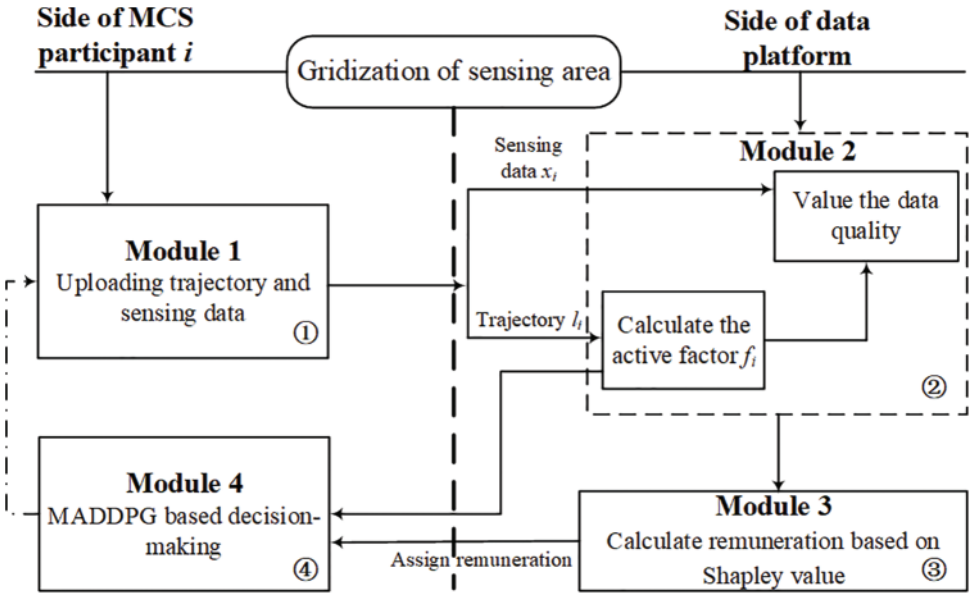

Figure 1: Flowchart of the proposed MARCS framework

MARCS framework models the interaction between MCS participants and the platform as a partially observable Markov game (POMG). In the MCS participant side, each agent/participant as a non-cooperative player aims to maximize her long-term return earned from the platform, by strategically selecting the sensing action/effort to report data in each round. On the platform side, practically, rather than directly observing the participants' actions, the platform can observe the data trajectories of participants, and the reported data, while the platform should appropriately remunerate certain agents to incentivize them to provide more high-quality data to optimize its overall objective.

The four modules shown in Fig. 1 are briefly described as follows:

- Module 1: At each sensing round t, each participant i uploads the movement trajectory l_i^{t-th} and sensing data x_i^{t-th} to platform. The trajectory l_i^{t-th} is characterized as a vector with size d, described in the following Section 3.1. The sensing data x_i^{t-th} is related to the effort/action a_i^{t-th} exerted by the participant.
- Module 2: Based on the participants' trajectories, the MCS platform first calculates the active factor for each participant's trajectory, f_i^{t-th}, which, in sense, is an indicator of the participant's trajectory quality to the platform. Then, the platform accumulates the sensing data x_i^{t-th} weighted with f_i^{t-th} as the overall value of MCS data, which corresponds to data quality. That is, the overall objective of data platform depends upon all agents' mobility trajectories and their reported data.
- Module 3: Through utilizing Shapley value theory, the platform infers the payment remunerated to the participant i, p_i^{t-th}, which accommodates various coalitional structures among agents, and implicitly characterizes the relationships among MCS participants without prior knowledge.
- Module 4: Using MADDPG algorithm, each agent dynamically adjusts the sensing action of the next round based on the received remuneration, and the observed environment state, including the active factor of the participant's trajectory, and the action history of her own.

The following subsections present the detailed procedures of all these four modules:

3.1 Module 1 on the MCS Participant Side: Uploading Trajectory and Sensing Data by Each Participant

Consider an MCS campaign over a finite and discrete-time horizon $T = \{1, \ldots, \mathrm{T}\}$, and N voluntary MCS participants, equipped with sensors perform sensing tasks for the data platform. At each timeslot/sensing round $t \in T$, the trajectory data and sensing data are uploaded to the platform. Each participant's trajectory in the t-th sensing round is discretized with a fixed time interval, e.g., 20 min in our experiments, into d samples. That is, for the t-th timeslot, there exist d trajectory time periods $T^{t-th} = \{t_1^{t-th}, t_2^{t-th}, \ldots, t_d^{t-th}\}$, where t_j^{t-th} represents the j-th sampling time on mobility trajectory in the t-th time slot. Then, the movement trajectory l_i^{t-th} of participant u_i in the timeslot t can be characterized as a vector with size d, shown as Eq. (1).

$$l_i^{t-th} = \{l_{i,1}^{t-th}, l_{i,2}^{t-th}, \ldots, l_{i,d}^{t-th}\} \tag{1}$$

where $l_{i,j}^{t-th}$ represents a specific grid in sensing area R that the participant u_i visits at the j-th sampled trajectory time in the t-th time slot, $i \in U$, $j \in [1, d]$, $l_{i,j}^{t-th} \in R$.

The sensing data x_i^{t-th} reported by the participant u_i depends on the chosen effort level a_i^{t-th} at the t-th time slot, which can be characterized as the participant's productive function P_i, shown as the Eq. (2).

$$x_i^{t-th} = P_i\left(a_i^{t-th}\right) \tag{2}$$

Note that $P_i(\cdot)$ can be any concave function, without loss of generality, in our experiments, it is simply set as a linear function with constant parameter.

3.2 Module 2 on the Platform Side: Calculating the Active Factor for Each Participant's Trajectory

To the platform, the overall value of all uploaded MCS data is related to two aspects: the active factors of participants' trajectories and reported sensing data by participants. The former is the indicator of the participant's trajectory quality, and the latter indicates the quality of sensing data.

Definition 1: Activity factor of participant's trajectory

On the platform side, all participants' trajectories l_U^{t-th} in the t-th time slot can be characterized as a $|U| \times d$ matrix, given as Eq. (3).

$$L_U^{t-th} = \begin{bmatrix} l_{11}^{t-th} & \cdots & l_{1d}^{t-th} \\ \vdots & \ddots & \vdots \\ l_{|U|1}^{t-th} & \cdots & l_{|U|d}^{t-th} \end{bmatrix} \tag{3}$$

where the i-th row denotes the sampled trajectory vector of the participant u_i at the timeslot t.

Using the trajectory matrix L_U^{t-th} of participants in the t-th timeslot, all participant's visiting-or-not matrix (VoN) V_U^{t-th} with size $|U| \times (m+n)$, and matrix of visiting times to hotspots (VTH) H_U^{t-th} with size $|U| \times n$ can be respectively obtained, shown as Eqs. (4) and (5).

$$V_U^{t-th} = \begin{bmatrix} v_{11}^{t-th} & \cdots v_{1m}^{t-th} v_{1(m+1)}^{t-th} \cdots & v_{1(m+n)}^{t-th} \\ \vdots & \vdots & \vdots \\ v_{|U|1}^{t-th} & \cdots v_{|U|m}^{t-th} v_{|U|(m+1)}^{t-th} \cdots & v_{|U|(m+n)}^{t-th} \end{bmatrix} \tag{4}$$

where each entry v_{ij}^{t-th} in VoN matrix V_U^{t-th} with binary value 0 or 1 characterizes whether or not the specific participant u_i has visited the specific grid j in the timeslot. Note that $j \in \{1, \ldots, m\}$ denotes the common grids, and $j \in \{(m+1), \ldots, (m+n)\}$ represents the hotspot grids.

$$H_U^{t-th} = \begin{bmatrix} h_{11}^{t-th} & \cdots & h_{1n}^{t-th} \\ \vdots & \ddots & \vdots \\ h_{|U|1}^{t-th} & \cdots & h_{|U|n}^{t-th} \end{bmatrix} \tag{5}$$

In VTH matrix H_U^{t-th}, each entry h_{ij}^{t-th} records the number of times that the specific participant u_i has visited the specific hotspot grid j.

Intuitively, besides covering all the grids with MCS data, hotspots should be paid more attention to and sensed more frequently. Thus, an integrated indicator measuring the quality of geospatial crowdsensing, so-called sensing coverage quality is proposed as Eq. (6), which is composed of the ratio of covering the whole sensing area (CP), and the degree of covering hotspots (CD).

$$F_U^{t-th} = \alpha \times CP_U^{t-th} + \beta \times log_{10}[CD_U^{t-th}] \tag{6}$$

where the weights α and β are used to adjust the proportion of CP and CD that is used in computing coverage quality.

The coverage ratio CP_U^{t-th}, inferred from the VoN matrix V_U^{t-th}, can be calculated as the ratio of the number of grids visited by all participants to the number of all grids in sensing area, shown as Eq. (7).

$$CP_U^{t-th} = \frac{\sum_{i=1}^{|U|}(\sum_{j=1}^{m} v_{ij}^{t-th} \times W_g + \sum_{j=m+1}^{m+n} v_{ij}^{t-th} \times W_h)}{m+n} \tag{7}$$

where the weights W_h and W_g are used to characterize the different impacts of common grids and hotspots on the coverage ratio.

The coverage degree of hotspots CD_U^{t-th} is calculated as the average cumulative times that all participants in set U have visited hotspots in the t-th timeslot, shown in Eq. (8).

$$CD_U^{t-th} = \frac{\sum_{i=1}^{|U|}\sum_{j=1}^{n} h_{ij}^{t-th}}{n} \tag{8}$$

Then, the actor factor of each participant's trajectory f_i^{t-th} can be defined as Eq. (9), which measures the participant's activity: the larger the $f_i^{t-th}(U)$ is, the more active the participant u_i is.

$$f_i^{t-th}(U) = F_U^{t-th} - F_{U|i}^{t-th} \tag{9}$$

where $F_{U|i}^{t-th}$ represents the coverage quality provided by other participants after removing the participant u_i.

Finally, through combing and accumulating the active factors and sense data of all participants, the overall data value, i.e., the platform's benefit is represented as Eq. (10).

$$B_U^{t-th} = \sum_{i=1}^{|U|} x_i^{t-th} \times f_i^{t-th}(U) \tag{10}$$

In brief, the platform's objective is to maximize the accumulative MCS data value by taking into account both the coverage quality of participants' trajectories and sensing data quality.

3.3 Module 3 on the Platform Side: Shapley Value-Based Remuneration to MCS Participants

Sensing data is costly for agents; therefore, it is necessary for the platform to appropriately compensate MCS participants for providing high-quality data. By considering different participants set as coalition structures, and the overall data values of coalitions as a global benefit, Shapley value, a tool from game theory offers a rigorous, intuitive, and axiomatic way to distribute the collective profit of the platform across individual agents.

Technically, the remuneration $p_i^{t-th}(U)$ assigned to the participant u_i in the t-th time slot is given as the Eq. (11).

$$p_i^{t-th}(U) = \sum_{S \subseteq U} \frac{[(|S|-1)!(|U|-|S|)!]}{|U|!} \times [B_S^{t-th} - B_{S|i}^{t-th}] \tag{11}$$

In summary, Shapley value assigns payment to an individual agent u_i by computing a weighted average of the benefit increase when u_i works with coalition S vs. when u_i does not work with S (i.e., a quantity known as the marginal contribution of the agent u_i). Through averaging this difference over all possible coalitions, to which the agent u_i does not belong, i.e., $S|i$, the data Shapley value of each agent served as the remuneration to the agent can be obtained. Shapley value is a unique and fair way to quantify the importance of each agent's sensing data and satisfies the properties of efficiency, symmetry, nullity, linearity, and coherency. Since Shapley value takes all kinds of participant coalitions into account, the implicit relations among all agents can be in sufficient consideration without prior knowledge, which further facilitates the learning of local agents.

Note that as shown in Eq. (11), for $|U| = n$ participants, calculating the exact Shapley value for a specific participant requires averaging over 2^{n-1} possible coalitions, which is computationally expensive, particularly when the Shapley values for all participants are considered. Therefore, this involves computing $n(2^{n-1})$ coalitions in total to obtain values for all players. While this value can be reduced to 2^n coalitions by reusing calculations for shared coalitions across players, the cost remains exponential with respect to the number of participants.

To address the computational challenge of calculating Shapley values, an effective method is to approximate the Shapley value using Monte Carlo sampling [29], instead of calculating exact contributions for 2^{n-1} coalitions per participant, this method randomly samples coalitions and computes the marginal contribution of a participant by comparing the outcomes with and without their inclusion in these coalitions. The Shapley value is then estimated as the average of these marginal contributions over multiple samples. This approach significantly reduces the computational cost while maintaining reasonable accuracy, making it suitable for large-scale systems with many participants. Besides, in literature, there exist various approximation methods to estimate the Shapley value, including truncated Monte Carlo and Gradient Shapley, among others [30,31].

3.4 Module 4 on the Participant Side: MADDPG-Based Decision-Making by Each MCS Participant

To participate in a geospatial MCS task, each user u_i needs to select an action a_i^{t-th} and sends sensed data to the data platform in time slot t. Performing sensing tasks will incur costs for participant, and meanwhile, as described above, each user u_i can receive the remuneration $p_i^{t-th}(U)$ from data platform. Assuming that each participant's utility exhibits additive separability in earned payment and cost of action, then the utility (i.e., income) of the participant u_i, r_i^{t-th} can be calculated with Eq. (12).

$$r_i^{t-th} = p_i^{t-th}(U) - c_i \cdot a_i^{t-th} = p_i^{t-th}(U) - c_i \cdot P_i^{-1}(x_i) \tag{12}$$

where $c_i \geq 0$ is constant and, and $P_i^{-1}(\cdot)$ is the inverse productive function of the participant u_i only known to the participant.

The goal of our work is to find a continuous decision a_i^{t-th} that maximizes the income $\sum_{t=1}^{T} \gamma^t r_i^{t-th}$ of participant u_i, where T is the total number of time slots and $0 < \gamma \leq 1$ is a predefined discount factor. In our work, from the perspective of MCS participants, the non-cooperative multi-agent sequential decision-making in MCS systems with N agents is modeled as a partially observable Markov Games (POMG), generalizing MDPs to multiple agents that simultaneously interact within a shared environment and possibly with each other. POMG is mathematically represented by the tuple $(U, \mathcal{S}, \{\mathcal{O}_i\}, \{\mathcal{A}_i\}, \{\mathcal{A}\}, \mathcal{P}, \{\mathcal{R}_i\}, \gamma)$, where $U = \{u_1, u_2, \ldots, u_N\}$ denotes the set of agents, $\mathcal{S}$ is the set of global environment states; $\mathcal{O}_i$ is the set of states locally observed by the participant u_i, and $\mathcal{S} = \mathcal{O}_1 \times \ldots \times \mathcal{O}_{|U|}$; $\mathcal{A}_i$ is the set of individual action space of the participant u_i, $\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_{|U|}$ is the joint action space of all agents. The transition probability function is denoted by $\mathcal{P}$; the reward function $\mathcal{R}_i$ is associated with the agent u_i; and the discount factor is γ.

In POMG, each participant u_i can't observe the whole state space but merely a subset $\mathcal{O}_i \subseteq \mathcal{S}$. In our scheme, $\mathcal{O}_i = \{f_i^1, f_i^2, \ldots, f_i^T\}$, implies that each participant can only observe her own active factors of trajectories as the system state. At sensing round t, each agent u_i selects and executes an action depending on the individual policy $\pi_i^t: \{\mathcal{O}_i\}_{t>0} \rightarrow a_i^t$. Note that participant does not know others' actions $\left\{a_j^t, j \in U, j \neq i\right\}$, since each agent is making decision independently. Each agent chooses an action simultaneously to formulate a joint action $a^t = (a_1^t, a_2^t, \ldots, a_{|U|}^t)$. The MCS system evolves from state $s^t \in \mathcal{S}$ under the joint actionwith respect to the transition probability function $\mathcal{P}$ to the next state $s^{t+1} \in \mathcal{S}$, while each agent receives r_i^t as immediate reward. In brief, for the MCS system, each step's state is represented as $\mathcal{O}_1 \times \ldots \times \mathcal{O}_{|U|}$. By taking a joint action for all MCS participants $\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_{|U|}$, two functions related to state and actions are defined: (a) the sensing environment takes a state transition function: $\mathcal{P}: \mathcal{O}_1 \times \ldots \times \mathcal{O}_{|U|} \times \mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_{|U|} \rightarrow \mathcal{O}_1 \times \ldots \times \mathcal{O}_{|U|}$, and (b) the reward for each agent u_i: $\mathcal{O}_1 \times \ldots \times \mathcal{O}_{|U|} \times \mathcal{A}_i \rightarrow \mathcal{R}_i$.

In multi-agent POMG, there exist the following two challenges: non-stationary participants' sensing environment, and the partially observable information (e.g., each participant i does not know others' actions, etc.). Our work adopts multiagent deep deterministic policy gradient (MADDPG) as the solution scheme, which features the centralized training and decentralized execution paradigm (CDTE) [32]. CTDE reconciles the trade-off between independent and centralized approaches by allowing agents to exchange additional information during training while maintaining scalability and addressing non-stationarity. Specifically, centralized training entails policies being updated based on mutual information exchange among agents, leveraging global observations or actions. Decentralized execution ensures that each agent determines its actions independently based solely on its local observations. This paradigm enables the Critic to utilize extra information, such as the actions or observations of other agents, during training to facilitate learning, while the Actors make decisions autonomously during execution, addressing the partially observable nature of the environment and improving performance.

Fig. 2 illustrates the multi-agent decentralized Actor and centralized Critic components of MARCS, where both Actors and Critics are trained in a centralized way, and only the trained Actors are used during the execution phase. As shown in Fig. 2, during training, first, all local agents interact with the environment and take actions according to their observations and history. Then, using these actions and observations as inputs, Critic of each agent is trained to estimate the action value. That is actions from other agents can be added to help each agent's Critic learn the interactions among different agents. The advantage of centralized training lies in that, since Critic has the full observation of all participants' actions during the training process, it faces a stationary environment, and avoids the non-stationary environment caused by other participants' actions and interactions.

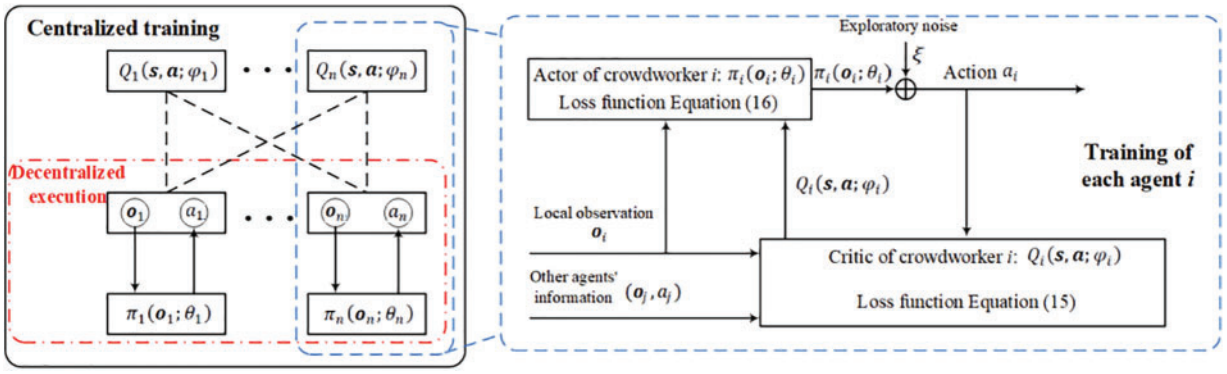

Figure 2: Illustration of multi-agent decentralized actors and centralized Critic components of MARCS

Once trained, the Actor does not need information from the Critic network, and can directly output actions. That is, during real implementation, each participant can only use state locally observed to make decisions.

MADDPG equips each agent u_i with four neural networks: two for Actor, and two for Critic. The former two neural networks correspond to an Actor's policy network $\pi_i(o_i; \theta_i)$ and an evaluation network $\pi_i^{'}(o_i^{'}; \theta_i^{'})$, both respectively parameterized by θ_i and $\theta_i^{'}$. Note that $o_i, o_i^{'} \in \mathcal{O}_i$, represent the locally observed state by the participant. The latter two neural networks are Critic network $Q_i(s, a; \varphi_i)$ and its target network $Q_i^{'}(s', a'; \varphi_i^{'})$, similarly both respectively parameterized by φ_i and $\varphi_i^{'}$. Note that $s, s' \in \mathcal{S}$, and $a, a' \in \mathcal{A}$ denotes that, in the training, the Critic can observe the global states and actions.

The algorithmic process for the t-th timeslot is described as follows. For clear presentation, the superscript t in notations is omitted. The MARCS first initializes the network parameters of the four neural networks mentioned above. The action of participant u_i is represented as Eq. (13).

$$a_i = \pi_i\left(o_i; \theta_i\right) + \xi \tag{13}$$

where o_i represents the historical activity factors of the participant u_i. Since each agent's action set is continuous, an exploratory noise is added to the deterministic action.

At the global environment/system state of $s \in \mathcal{S}$, each agent chooses and executes an action simultaneously, denoted the joint actions $a = (a_1, a_2, \ldots, a_{|U|})$. Each agent receives the payment and correspondingly calculate the reward r_i, denoted the joint rewards $r = (r_1, r_2, \ldots, r_{|U|})$. The environment state changed into $s' \in \mathcal{S}$. Then, the experience (s, a, r, s') is stored in the experience replay memory D.

To update the network parameters, for each participant, the number of K experiences $(s^j, a^j, r^j, s^{'j})$ are sampled from the experience replay memory D, where the superscript j represents the j-th experience. Then, the Critic's target network is used to calculate the expected return y^j for each experience with Eq. (14).

$$y^j = r_i^j + \gamma Q_i^{'}\left(s^{'j}, a_1^{'}, a_2^{'}, \ldots, a_{|U|}^{'}\Big|_{k=\{1,2,\ldots N\}}^{a_k^{'}=\pi_k\left(o_k^{'}; \theta_k^{'}\right)}\right) \tag{14}$$

where γ is the discount factor. Then the method of minimizing losses is used to update the Critic network parameters with Eq. (15).

$$\nabla_{\varphi_i} J(\varphi_i) = \frac{1}{K}\sum_{j=1}^{K}\left(y^j - Q_i\left(s^j, a^j; \varphi_i\right)\right)^2 \tag{15}$$

Then Actor's policy network parameters are updated with Eq. (16).

$$\nabla_{\theta_i} J(\theta_i) = \frac{1}{K} \sum_{j=1}^{K} \nabla_{\theta_i} \pi(o_i^j; \theta_i) \nabla_{a_i} Q_i(s^j, a_1^j, a_2^j, \ldots, a_{|U|}^j; \varphi_i) \tag{16}$$

After certain number of steps is executed, the parameters of the evaluation network of Actors and Critics of all participants are updated using Eqs. (17) and (18).

$$\theta_i^{'} = \tau \cdot \theta_i + (1 - \tau) \cdot \theta_i^{'} \tag{17}$$

$$\varphi_i^{'} = \tau \cdot \varphi_i + (1 - \tau) \cdot \varphi_i^{'} \tag{18}$$

4 Mobility Trajectory Description and Experimental Results

In this section, we conduct comprehensive experiments based on the user's real trajectory dataset to verify the superiority of the proposed MARCS framework and compare it with other state-of-the-art (SOTA) benchmark schemes.

4.1 Experimental Data Set and Settings

The experiments are carried out on a real dataset of participants' mobility trajectories, GeoLife dataset composed of trajectories of 182 users over five years, i.e., from April 2007 to August 2012, collected by (Microsoft Research Asia) Living Earth project. In detail, this dataset contains a total of 17,621 trajectories amounting toa total distance of about 1.2 million kilometers and a total duration of 50,176 h. Each trajectory is represented as a record consisting of a time-stamped location with a specific latitude, longitude, and altitude.

Considering the distribution of participants' trajectories in the GeoLife dataset is skewed, a region with the densest users' trajectories is chosen as the MCS sensing area. In our experiments, shown as a red dotted area in Fig. 3, the square within the eastern longitude from 116.31 to 116.35, and the northern latitude from 39.975 to 40.02, i.e., part of Haidian District, Beijing, is used as the sensing area. Then, the sensing region is further divided into 20 * 20 grids, i.e., a total of 400 grids of the same size, and among them, 20 grids are evenly and randomly chosen as the hotspots.

The total number of 100 participants' trajectories within the period of the same 40 days are randomly extracted from the GeoLife dataset, and used as the voluntary MCS participants and their mobile trajectories in our experiments. Then these extracted trajectories are further split into four test cycles: each cycle includes 10 days, and each day is intentionally divided into 2 timeslots. The range of the first timeslot is from 0:00 to 8:00, and the second range of the second time is from 8:00 to 20:00. In other words, each test cycle totally includes 20 timeslots.

The accumulative incomes of participants (shown as Eq. (12)) and the platform's benefit (given as Eq. (10)) in test cycles are used as metrics to demonstrate the performance of our proposed MARCS framework by comparing it with other benchmark schemes.

4.2 Baseline Schemes

In our evaluation, three methods are used as baselines for comparison with our proposed MARCS scheme, including Model Predictive Control (MPC) [33,34], and two deep reinforcement learning algorithms for multiple independent agents: Actor-Critic and DDPG.

MPC method is a standard method used for resource allocation and optimal control, in which a local model for the reward dynamics is linearly fit for each participant and the MPC is solved under a fixed number of timeslots to obtain the predicted optimal sensing action for each timeslot.

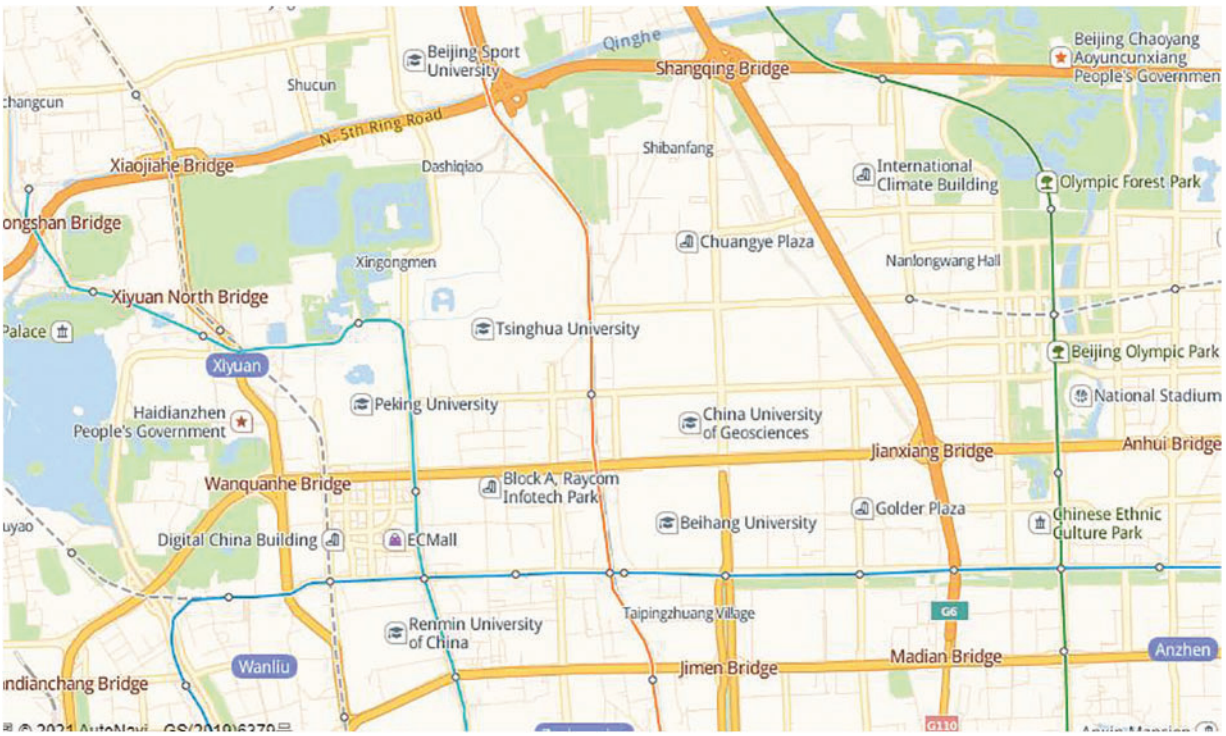

Figure 3: Screenshot of the MCS sensing region

Actor-Critic (AC) is a method based on the idea of policy gradient and consists of both Actor neural network and Critic neural network. The former selects behavior based on probability, and the latter evaluates the actor's behavior score. Actor modifies the behavior probability according to critic's evaluation.

DDPG also follows the Actor and Critic structure, but is an off-policy method. It aims to select the optimal action in continuous actions to maximize the agent's long-term return. It can be well applied to decision-making in continuous action space, and efficiently utilize the experience playback mechanism and target network in deep Q-learning network to improve the stability of the algorithm.

Note that in MARCS, DDPG, and AC methods, for fair comparison, both the policy network and the target network are fully connected neural networks with 2 hidden layers and 64 hidden units.

4.3 Experimental Results

In this section, performance analysis was conducted respectively from the participant side and platform side.

- *1) Participant side*

Fig. 4 shows the cumulative incomes of all participants respectively for these four schemes when the number of voluntary participants varies from 10, 20, 30, and 40. Taking 10 participants as an example, the MARCS scheme can achieve an average cumulative benefit of 447.33 in one cycle, which is the average result of 10 runs in 4 test cycles. For the accuracy of the experiment, the maximum and minimum values of these 40 experiments are also given.

From Fig. 4, we can intuitively observe that the revenue of MARCS is superior to other benchmark schemes, with almost 35%, 53%, and 100% higher than DDPG, Actor-Critic, and MPC, respectively. These

all indicate that the MARCS scheme has superior performance on the participant side compared to other schemes.

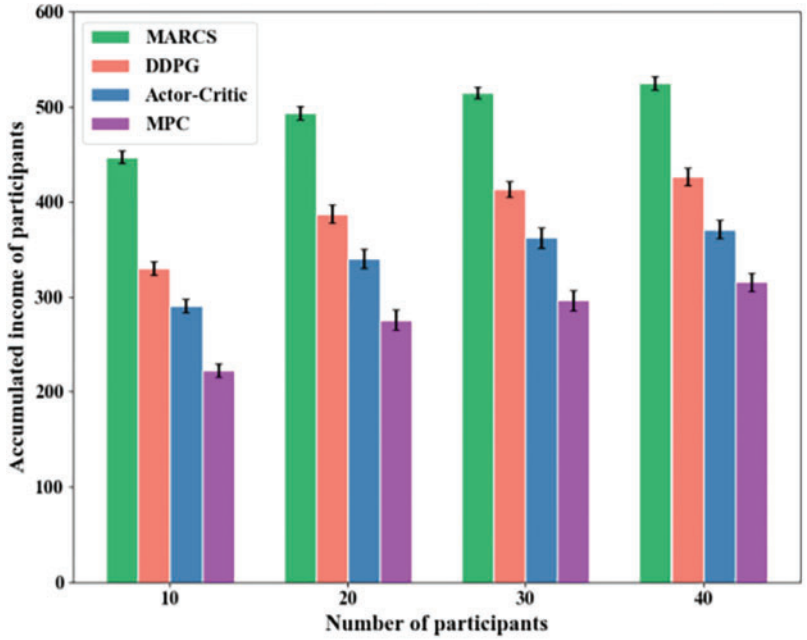

Figure 4: Comparison of cumulative rewards of participants under different numbers of voluntary participants

The reason for the superiority is straightforward. MARCS uses centralized training, which has full observation of the environment state and actions of all participants and facilitates criticalto efficient learning. While AC and DDPG just train an independent Actor-Critic learner to learn its own sensing decision. MPC agent performs worst, due to the fact that the linear prediction model in MPC can't find a good representation of system dynamics at all.

Fig. 5 shows the income of participants at each time slot during a testing cycle. Overall, compared to other schemes, the MARCS scheme also achieved optimal performance at the fine level of time slot. As the time slots go by (i.e., the learning step increases), the income trend of participants in each scheme shifts from upward to flat.

- *2) MCS Platform side*

This section considers the performance of each scheme from the MCS data platform side, in terms of both the overall performance and performance of timeslot.

Fig. 6 shows the comparison of platform benefits under the different numbers of voluntary participants, in terms of the average of 10 experiments over 4 cycles. It can be seen that, in all scenarios, MARCS achieves the best performance.

Fig. 7 shows the platform's benefit at each time slot during a testing cycle. It can be observed that as the learning step unfolds, the platform's benefit correspondingly increases since using data Shapely as remuneration to each participant can align the platform's overall benefit with the agent's utility, and lead to some agents with high trajectory active factors can gradually learn to exert sufficient effort to maximize her long-term return, which in turn increases the platform's overall benefit.

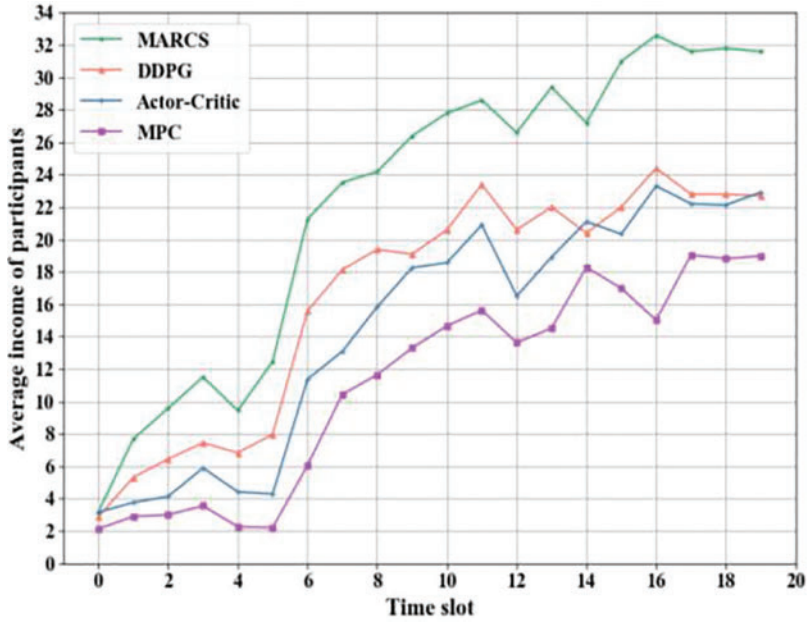

Figure 5: Comparison of the average income of participants varying with different time slots in a test cycle

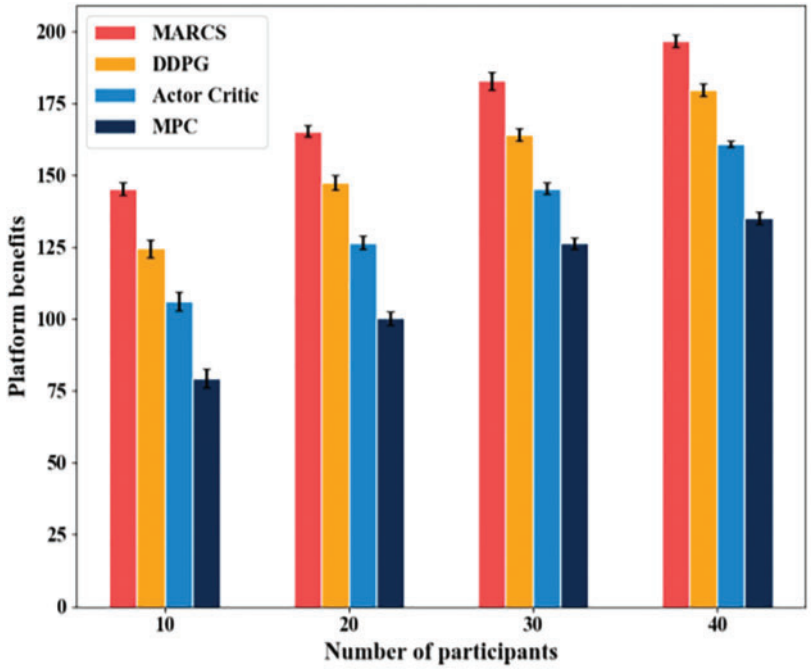

Figure 6: Comparison of platform benefits varying with the number of selected participants

4.4 Complexity Analysis

As Table 1 indicates, for n participants, the algorithm complexity of MADDPG can be represented as $O(n)$. Furthermore, the computational cost of MADDPG is influenced by multiple factors, including the dimensionality of the state space, the dimensionality of the action space, the number of neurons in the hidden layers of the neural network, and so on [35]. Since the number of subsets of the set of n participants is 2^n, evaluating the Shapley value for player i has time complexity $O(2^n)$. This exponential complexity poses

challenges for large-scale scenarios. As illustrated in Section 3.3, approximate estimation methods such as the truncated Monte Carlo approach can be utilized to mitigate this issue, reducing the complexity to a more manageable level. It is important to note that there is often a tradeoff between computational complexity and estimation accuracy. In practice, the choice of estimation method and its parameters should be carefully adjusted based on the specific requirements and constraints of the scenario.

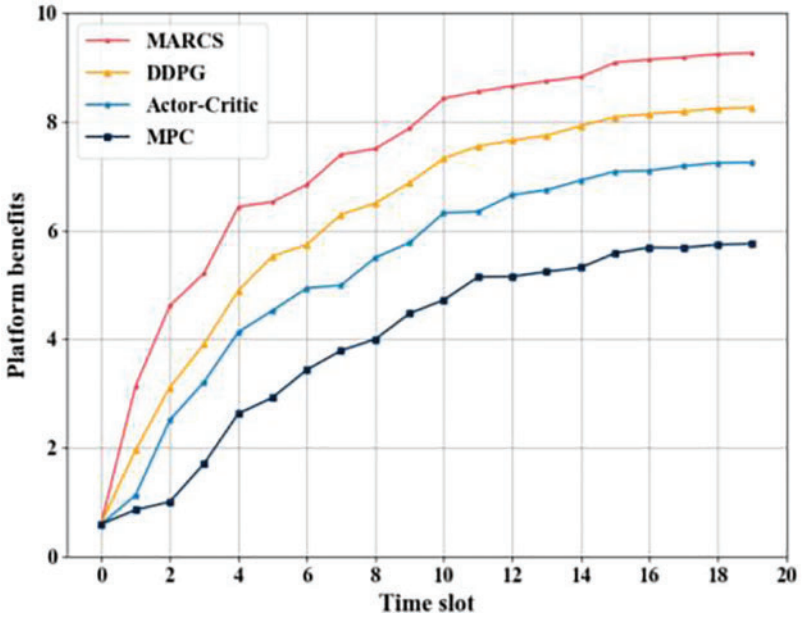

Figure 7: Comparison of platform benefits varying with different time slots in a test cycle

Table 1: Algorithm complexity analysis

	MADDPG	Shapley value calculation
Time complexity (n agents)	$O(n)$	$O(2^n)$

5 Conclusion

In opportunistic geospatial MCS tasks, the interests of mobile participants and data platforms should be both accommodated and aligned with each other. Practically, each participant aims to maximize her long-term return, by strategically determining the action in each sensing round, while each participant's decision faces many uncertainties, such as the participants can't observe the others' actions, and the trajectories of participants' many have implicitly complex inter-dependence to the platform's overall benefit. To address the above issue, in our work, by treating each participant as a learning agent, a multiple-agent DRL-based MCS geospatial crowdsensing framework, MARCS is proposed. Specifically, the data platform integrates the active factor of each participant's trajectory and her sensing data as the overall objective and utilizes data Shapley value correspondingly remunerates each participant according to her marginal contribution to various coalitional structures of agents. On the MCS participant side, treating the remuneration minus the sensing cost as an immediate reward in each sensing round, each agent adopts CTDE-based MADDPG to learn the strategic action based on the locally observed environment state, to maximize her long-term return. Experimental results on real mobility trajectory datasets show that MARCS outperforms DDPG,

Actor-Critic, and MPC, with participant-side revenue increases of approximately 35%, 53%, and 100%, respectively, and similar improvements observed on the platform side. In the future, the testing could be expanded to larger and more diverse datasets to further evaluate the scalability and robustness of MARCS. Additionally, exploring the integration of MARCS with other advanced reinforcement learning techniques and incorporating real-time dynamic environments could further enhance its performance and applicability in practical MCS scenarios.

Acknowledgement: The authors would like to thank the anonymous reviewers and editors for their valuable comments, which helped improve the quality of the paper greatly.

Funding Statement: This research is sponsored by Qinglan Project of Jiangsu Province, and Jiangsu Provincial Key Research and Development Program (No. BE2020084-1).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Yiqin Wang, Yufeng Wang; data collection: Yiqin Wang; analysis and interpretation of results: Jianhua Ma, Qun Jin; draft manuscript preparation: Yiqin Wang, Yufeng Wang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Nomenclature

l_i^{t-th}	The movement trajectory of participant i in time slot t
a_i^{t-th}, x_i^{t-th}	Action/effort of participant i at a certain time slot t, and the sensing data in corresponding action
m, n	The number of common grids and hotspot grids in sensing region R
F_U^{t-th}	Coverage quality of |U| participants' trajectories in time slot t
CP_U^{t-th}	The ratio between the number of grids visited by |U| participants and the number of all grids
CD_U^{t-th}	The average cumulative times that participants in U visit hotspot grids in time slot t
$f_i^{t-th}(U)$	Active factor of participant i's trajectory
B_U^{t-th}	The value of MCS data sensed by the participant set U to platform at the t-th timeslot
p_i^{t-th}	Remuneration to the participant u_i by platform at the t-th timeslot
s^t	The whole MCS environmental state at the t-th timeslot
o_i^t	The MCS environmental state locally observed by the participant u_i

References

1. Yu Z, Ma H, Guo B, Yang Z. Crowdsensing 2. 0. Commun ACM. 2021;64(11):76–80. doi: 10.1145/3481621.
2. Guo B, Wan Y, Yen N, Huang R, Zhou X. Mobile crowd sensing and computing: the review of an emerging human-powered sensing paradigm. ACM Comput Surv. 2015;48(1):7.
3. Ma H, Zhao D, Yuan P. Opportunities in mobile crowd sensing. IEEE Commun Mag. 2014;52(8):29–35. doi: 10.1109/MCOM.2014.6871666.
4. Wu L, Xiong Y, Liu KZ, She J. A crowdsensing market based on game theory: Participant incentive, task assignment and pricing guidance. Int J Commun Netw Distrib Syst. 2022;28(5):517–33.
5. Xu J, Zhou Y, Ding Y, Yang D, Xu L. Biobjective robust incentive mechanism design for mobile crowdsensing. IEEE Internet Things J. 2021;8(19):14971–84. doi: 10.1109/JIOT.2021.3072953.

6. Zhang J, Zhang Y, Wu H, Li W. An ordered submodularity-based budget-feasible mechanism for opportunistic mobile crowdsensing task allocation and pricing. IEEE Trans Mobile Comput. 2024;23(2):1278–94. doi: 10.1109/TMC.2022.3232513.
7. Wang X, Wang S, Liang X, Zhao D, Huang J, Xu X, et al. Deep reinforcement learning: a survey. IEEE Trans Neural Netw Learn Syst. 2024;35(4):5064–78. doi: 10.1109/TNNLS.2022.3207346.
8. Xiao L, Li Y, Han G, Dai H, Poor HV. A secure mobile crowdsensing game with deep reinforcement learning. IEEE Trans Inf Forens Secur. 2018;13(1):35–47. doi: 10.1109/TIFS.10206.
9. Gronauer S, Diepold K. Multi-agent deep reinforcement learning: A survey. Artif Intell Rev. 2022;55:895–943. doi: 10.1007/s10462-021-09996-w.
10. Nguyen TT, Nguyen ND, Nahavandi S. Deep reinforcement learning for multi-agent systems: a review of challenges, solutions and applications. IEEE Trans Cybern. 2020;50(9):3826–39. doi: 10.1109/TCYB.6221036.
11. Zhang X, Yang Z, Zhou Z, Cai H, Chen L, Li X. Free market of crowdsourcing: incentive mechanism design for mobile sensing. IEEE Trans Parall Distrib Syst. 2014;25(12):3190–200. doi: 10.1109/TPDS.2013.2297112.
12. Jiang C, Lin G, Lin D, Huang J. Scalable mobile crowdsensing via peer-to-peer data sharing. IEEE Trans Mobile Comput. 2018;17(4):898–912. doi: 10.1109/TMC.2017.2743718.
13. Cao B, Xia S, Han J, Li Y. A distributed game methodology for crowdsensing in uncertain wireless scenario. IEEE Trans Mobile Comput. 2020;19(1):15–28. doi: 10.1109/TMC.7755.
14. Zhan Y, Xia Y, Zhang J. Incentive mechanism in platform-centric mobile crowdsensing: a one-to-many bargaining approach. Comput Netw. 2018;132:40–52. doi: 10.1016/j.comnet.2017.12.013.
15. Zhan Y, Xia Y, Zhang J, Wang Y. Incentive mechanism design in mobile opportunistic data collection with time sensitivity. IEEE Internet Things J. 2018;5(1):246–56. doi: 10.1109/JIOT.2017.2779176.
16. Guo B, Liu Y, Wu W, Yu Z, Han Q. ActiveCrowd: a framework for optimized multitask allocation in mobile crowdsensing systems. IEEE Trans Human-Mach Syst. 2017;47(3):392–403. doi: 10.1109/THMS.2016.2599489.
17. Li X, Zhang X. Multi-task allocation under time constraints in mobile crowdsensing. IEEE Trans Mobile Comput. 2021;20(4):1494–510. doi: 10.1109/TMC.7755.
18. Xiao L, Chen T, Xie C, Poor HV. Mobile crowdsensing games in vehicular networks. IEEE Trans Veh Technol. 2018;67(2):1535–45. doi: 10.1109/TVT.25.
19. Xu Y, Wang Y, Ma J, Jin Q. PSARE: a RL-based online participant selection scheme incorporating area coverage ratio and degree in mobile crowdsensing. IEEE Trans Veh Technol. 2022;71(10):10923–33. doi: 10.1109/TVT.2022.3183607.
20. Zhan Y, Xia Y, Zhang J, Li T, Wang Y. An incentive mechanism design for mobile crowdsensing with demand uncertainties. Inf Sci. 2020;528:1–16. doi: 10.1016/j.ins.2020.03.109.
21. Zhan Y, Liu CH, Zhao Y, Zhang J, Tang J. Free market of multi-leader multi-follower mobile crowdsensing: an incentive mechanism design by deep reinforcement learning. IEEE Trans Mobile Comput. 2020;19(10):2316–29. doi: 10.1109/TMC.7755.
22. Xu C, Song W. Intelligent task allocation for mobile crowdsensing with graph attention network and deep reinforcement learning. IEEE Trans Netw Sci Eng. 2023;10(2):1032–48. doi: 10.1109/TNSE.2022.3226422.
23. Chen Y, Wang H. IntelligentCrowd: mobile crowdsensing via multi-agent reinforcement learning. IEEE Trans Emerg Topics Comput Intell. 2021;5(5):840–5. doi: 10.1109/TETCI.2020.3042244.
24. Dongare S, Ortiz A, Klein A. Federated deep reinforcement learning for task participation in mobile crowdsensing. In: Proceeding of the IEEE Global Communications Conference (GLOBECOM); 2023; Kuala Lumpur, Malaysia. p. 4436–41.
25. Xu C, Song W. Decentralized task assignment for mobile crowd-sensing with multi-agent deep reinforcement learning. IEEE Internet Things J. 2023;10(18):16564–78. doi: 10.1109/JIOT.2023.3268846.
26. Wang J, Zhang Y, Kim TK, Gu Y. Shapley Q-value: a local reward approach to solve global reward games. Proc AAAI Conf Artif Intell. 2020;34:7285–92.
27. Leitner S, Wall F. Decision-facilitating information in hidden-action setups: an agent-based approach. J Econ Interact Coord. 2021;16:323–58. doi: 10.1007/s11403-020-00297-z.

28. Rozemberczki B, Watson L, Bayer P, Yang HT, Kiss O, Nilsson S, et al. The shapley value in machine learning. In: Proceeding of the Thirty-First International Joint Conference on Artificial Intelligence; 2023; Montreal, QC, Canada. p. 5572–79.
29. Heuillet A, Couthouis F, D'ıaz-Rodr´ıguez N. Collective explainable AI: explaining cooperative strategies and agent contribution in multiagent reinforcement learning with shapley values. IEEE Comput Intell Mag. 2022;17(1):59–71. doi: 10.1109/MCI.2021.3129959.
30. Chen H, Covert IC, Lundberg SM, Lee S. Algorithms to estimate Shapley value feature attributions. Nature Mach Intell. 2023;5:590–601. doi: 10.1038/s42256-023-00657-x.
31. Wang J, Zhang Y, Gu Y, Kim T-K. SHAQ: incorporating Shapley value theory into multi-agent Q-learning. Adv Neural Inf Process Syst. 2022;35:5941–54.
32. Lowe R, Wu Y, Tamar A, Harb J, Abbeel P, Mordatch I. Multiagent actor-critic for mixed cooperative-competitive environments. In: Proceeding of Advances in Neural Information Processing Systems; 2017; Long Beach, CA, USA. p. 6379–90.
33. Garriga JL, Soroush M. Model predictive control tuning methods: a review. Indus Eng Chem Res. 2010;49(8):3505–15. doi: 10.1021/ie900323c.
34. Morato MM, Felix MS. Data science and model predictive control: a survey of recent advances on data-driven MPC algorithms. J Process Control. 2024;144:103327. doi: 10.1016/j.jprocont.2024.103327.
35. Du J, Kong Z, Sun A, Kang J, Niyato D, Chu X, et al. MADDPG-based joint service placement and task offloading in MEC empowered air-ground integrated networks. IEEE Internet Things J. 2024;11(6):10600–15. doi: 10.1109/JIOT.2023.3326820.