Sensing Strategy Optimization for Satellite-Assisted Vehicular Crowdsensing in Non-Terrestrial Networks

Arbil Chakma, *Graduate Student Member, IEEE*, Jingrong Wang, *Member, IEEE*, Quang Nhat Le, *Member, IEEE*, Octavia A. Dobre, *Fellow, IEEE*, and Trung Q. Duong, *Fellow, IEEE*

***Abstract*—Vehicular crowdsensing (VCS) is a paradigm that exploits vehicle mobility, on-board sensing capabilities, and drivers' smartphone sensors to collect large-scale, distributed information to provide intelligent, location-based services. Satellite-assisted VCS architecture can complement terrestrial networks by enabling wide-area and infrastructure-independent data collection. Incentivizing vehicles to participate in satellite-assisted VCS campaign remains a major challenge due to associated sensing and communication costs, requiring each vehicle to optimize their sensing level to maximize their received reward. Moreover, unlike conventional assumptions where all vehicles participate simultaneously, practical VCS scenarios are asynchronous, as vehicle may start and complete sensing tasks at different times. To capture this realistic setting, we propose an asynchronous multi-agent proximal policy optimization (A-MAPPO) algorithm within a centralized training and decentralized execution (CTDE) framework to optimize the sensing strategies of individual vehicles in a satellite-assisted VCS setting. A dynamic social network effect among vehicles is also incorporated to encourage vehicle participation driven by social benefits. Extensive numerical experiments are conducted to evaluate the performance of the proposed approach, demonstrating that A-MAPPO achieves superior performance compared with MASAC, DQN, Greedy-Q, and Random baselines.**

***Index Terms*—Vehicular crowdsensing, multi-agent deep reinforcement learning, proximal policy optimization, asynchronous learning, non-terrestrial networks.**

I. Introduction

THE integration of vehicle on-board sensors, driver's smartphone sensors, and vehicle mobility has introduced a specialized form of mobile crowdsensing (MCS), known as vehicular crowdsensing (VCS) [1], [2], [3]. A typical VCS framework generally consists of two major components: (1) a group of vehicles participating in crowdsensing campaign to collect context-aware and location-dependent sensing data, and (2) a centralized cloud server that aggregates and analyze the collected sensing data to provide services such as traffi congestion monitoring [4], [5], surface condition assessment [6], precise map construction [7], [8], and environmental monitoring [9]. Typically, a set of sensing tasks are generated by the centralized server and then distributed among vehicles that participate in crowdsensing. Each vehicle selects an appropriate sensing strategy and collects sensing data using its on-board sensors, such as cameras, radar, and smart devices, including smartphones or tablets [10]. The collected data is then sent to the centralized server via a communication network such as fourth generation (4G), long-term evolution (LTE), and fifth generation (5G) wireless networks. Upon receiving the data, the server processes and analyzes the sensing data to deliver the intended location-based services.

The overall effectiveness of a VCS system critically depends on the number of participating vehicles and the amount and quality of their sensing contributions [3]. Motivating vehicles to actively engage in a VCS campaign remains a significant challenge, as participation incurs additional costs, including battery consumption due to sensing and computation. It is essential to design a robust incentive mechanism that encourages the vehicles to contribute while balancing the associated costs of the crowdsensing tasks. Such mechanisms can leverage monetary rewards [11], social incentives [3], [12], [13], or reputation-based systems [14] to promote active participation, ensure sufficient data coverage, and enhance the overall performance and reliability of the VCS system. A well-designed VCS can significantly advance urban transportation management by improving traffic efficiency, road safety, and overall driving experience [15].

In traditional VCS systems, the vehicles connect with the centralized server which processes all information and carry on decision-making, typically communicating through a roadside ground base station [9]. They typically rely on infrastructure-dependent architectures that are feasible primarily in urban environments with well-established network coverage. However, vehicles traveling along long stretches of national highways or remote regions often experience intermittent or complete loss of connectivity due to the absence of terrestrial base stations. Building additional infrastructure to provide services in those areas is not often economically viable, leading to the use of alternative solutions such as satellite internet.

The adoption of satellite links into VCS frameworks can extend coverage to wide-area and infrastructure-independent environments, ensuring continuous data collection and service availability for vehicles beyond the reach of terrestrial networks [12], [16]. However, satellite-assisted VCS faces

The work of Octavia A. Dobre was supported by Canada Research Chairs Program under Grant CRC-2022-00187. This work was supported in part by Canada Excellence Research Chair Program under Grant CERC-2022-00109 and in part by the Natural Sciences and Engineering Research Council of Canada(NSERC)Discovery Grant Program under Grant RGPIN-2025-04941. Recommended for acceptance by Dr. Xianbin Wang.
(Corresponding author: Trung Q. Duong.)

The authors are with the Department of Electrical and Computer Engineering, Memorial University, St. John's NL A1B 3X9, Canada (e-mail: arbilc@mun.ca; jingrongw@mun.ca; qnle@mun.ca; odobre@mun.ca; tduong@mun.ca).

practical challenges, including higher communication latency, limited spectrum resources, and increased transmission costs compared with terrestrial networks. Nevertheless, the extensive coverage of the satellite link provides reliable support for highly mobile vehicles and dynamic crowdsensing, where frequent handovers and intermittent link availability are common. Overall, satellite links can complement terrestrial network by enhancing coverage robustness and service continuity.

Most existing VCS incentive mechanisms implicitly neglect communication cost [3], [15], thereby overlooking an important factor in practical VCS models. While such assumptions may be marginally acceptable in terrestrial VCS with short-range links, they become invalid in satellite-assisted VCS, since communication cost is significantly higher due to long propagation distance and severe path loss [17]. Directly applying ground-based incentive models in this context may lead to infeasible outcomes, where actual communication expense may exceed received rewards, ultimately discouraging participation of vehicles in the VCS game and risking system collapse. By explicitly modeling satellite communication costs within the incentive design, vehicles can rationally optimize their sensing level against their true operational cost, ensuring economically feasible participation and leading to a realistic system model.

Optimizing the sensing level is inherently a sequential decision-making problem, where each agent must consider resource consumption and network dynamics. While classical approaches such as brute-force and dynamic programming can provide optimal solutions in small or static settings [3], deep reinforcement learning (DRL) has proven itself as more effective, enabling agents to learn near-optimal sensing policies directly from interaction with the environment and demonstrating strong performance in real-time wireless communication scenarios.

In multi-agent DRL (MADRL) frameworks, it is commonly assumed that all agents operate synchronously, i.e., they start and terminate their tasks simultaneously [3], [15]. This assumption is primarily adopted for algorithmic simplicity and to simplify the training strategy. In VCS systems where vehicles independently decide their sensing actions, the problem can be naturally modeled as multi-agent system, with each vehicle acting as an independent agent. However, in practical VCS system, synchronization rarely holds since vehicles complete their tasks at different times due to task dependent completion condition or battery limitations. This asynchronicity introduces non-stationary and partially observability into learning process, which can significantly degrade the performance and stability of DRL-assisted sensing policies if not properly addressed.

In this paper, we propose a satellite-assisted VCS architecture that explicitly accounts for dynamic social effects among vehicles. An incentive mechanism is designed to jointly consider sensing costs and satellite-vehicle communication costs. To cope with the asynchronous nature of VCS and limited battery budget for sensing tasks, we develop an asynchronous MADRL approach to enable vehicles to adaptively determine their sensing policies under asynchronous participation. The main contributions of this proposed work are as follows:

- We formulate a non-cooperative VCS game between a satellite-deployed VCS server and participating vehicles in a non-terrestrial setting, where direct vehicle-to-satellite communication is considered, while taking into account their dynamic social relationships.
- We design an incentive mechanism that explicitly incorporates satellite-vehicle communication cost along with sensing reward, sensing cost, and social incentive to support efficient decision-making in satellite-assisted VCS environments.
- We propose an asynchronous multi-agent proximal policy optimization (A-MAPPO) algorithm in centralized training and decentralized execution (CTDE) manner to dynamically optimize sensing level of each vehicle in VCS environment.
- Extensive simulations are performed to evaluate the proposed algorithm's convergence behavior and performance gains relative to other baseline models.

A comparison of representative prior works in VCS is summarized in Table I. The remainder of this paper is organized as follows. First, related works are presented in Section II. Section III introduces system model and problem formulation, including detailed descriptions of the vehicle-satellite channel model, and social network model. An asynchronous MAPPO under the CTDE manner is proposed in the Section IV. Section V presents simulation result and analysis discussion. Finally, Section VI concludes the paper by summarizing the key findings.

II. Related Works

A. Vehicular Crowdsensing

Recently, VCS has received significant research attention due to its potential to improve road safety, traffic awareness, and overall driving experiences. Most of the existing studies focus on recruiting participant vehicles for VCS campaigns [18], [19], [20], [21], [27]. In particular, mobility-aware vehicle selection has been widely explored, where the key objective is to ensure that selected vehicles can complete their assigned sensing tasks within their residence time in the target area [18], [19]. Beyond mobility consideration, factors such as non-deterministic sensing tasks and uncertain mobile user behavior have also been studied [27]. While these studies provide valuable insights, most overlook the joint impact of roadside infrastructure sensing and communication limitations. Addressing this gap, an infrastructure-aware recruitment framework has been investigated [20] that selects the minimum set of vehicles capable of uploading fine-grained sensing data with wide and uniform spatial coverage. This work also accounts for heterogeneous sensing capabilities, dynamic traffic conditions, and payment budget constraints, offering a more holistic perspective on practical VCS deployment.

In parallel to vehicle selection, several studies focus on efficient task allocation among participating vehicles to enhance task completion, fairness, and privacy in VCS system. A long-term dynamic task assignment framework has been proposed to maximize sensing utility while jointly considering task urgency,

TABLE I
OVERVIEW OF PRIOR STUDIES AND THIS WORK

Studies	VCS	Satellite-assisted Architecture	Sensing Strategy Optimization	Incentive Design	Social Network	Asynchronous MADRL
[18], [19]	✓	×	×	×	×	×
[20], [21]	✓	×	✓	×	×	×
[12]	✓	✓	×	✓	×	×
[22]	✓	✓	✓	×	×	×
[23]–[27]	✓	×	×	✓	×	×
[3], [15]	✓	×	✓	✓	✓	×
This work	✓	✓	✓	✓	✓	✓

vehicle overload, and energy constraints, thereby ensuring both timely completion and fairness across sensing tasks [28]. Additionally, an intelligent MADRL-based approach has been explored, where human-like decision factors and incentive mechanism are integrated to dynamically prioritize crowdsensing tasks and overall system adaptability [15]. Furthermore, privacy-preserving task allocation mechanisms have gained attention, including methods that securely select vehicles based on their predicted future trajectories, enabling strong location privacy protection while still maximizing spatio-temporal sensing coverage [29].

The studies above have discussed on vehicle selection and task assignment, emphasizing the server platform perspective, focusing on vehicle recruitment, task allocation, data quality assurance, and system-level utility optimization. They often prioritize how to select vehicles or assign tasks to maximize coverage, reduce costs, or maintain sensing quality from a system design standpoint. However, they largely overlook the incentives or direct benefits provided to the participating vehicles or drivers. In practice, motivating drivers to contribute to VCS campaigns remains a significant challenge, as participation incurs sensing, and communication costs for vehicles with limited resources. To address this, several studies investigated incentive mechanisms explicitly linked to participation, enhancing coverage areas, and data quality [3], [11], [12], [13], [14], [23], [24], [25], [26], [30], [31].

Several studies have explored incentive mechanisms to encourage vehicle participation and improve both spatial [23] and temporal [24] sensing coverage in VCS systems. In both of the studies, the incentive mechanisms have been designed to motivate autonomous vehicles to deviate from their planned routes to enhance sensing coverage. Other works focus on dynamic pricing and incentive setting [25], [26]. To encourage sensing in areas with low vehicle presence, an incentive mechanism has been proposed to allocate higher payments to those areas derived though a Nash bargaining process that balances vehicle sensing costs with server utility [25]. Meanwhile, to ensure sensing quality within budget constraints, [26] jointly addresses vehicle selection and payment determination by considering bids, reputation, and coverage goals.

Beyond these efforts, other studies examine socially aware incentive effects, considering how social factors influence vehicle participation. For instance, authors in [3] have designed a socially aware incentive framework that enables each vehicle to optimize its sensing strategy to maximize individual utility. Building on this concept, augmented intelligence has been integrated with social factors to further enhance the optimization of vehicle sensing strategies [15].

These aforementioned studies are limited by their focus on terrestrial network infrastructure as vehicles heavily rely on roadside base stations to upload sensing data to centralized server. The emerging sixth generation (6G) network, with integrated satellite connectivity, has the potential to revolutionize VCS systems by enabling sensing and communication in areas with limited terrestrial coverage or during periods of network congestion. Research on integrating VCS with satellite communication remains largely unexplored, with only a few studies addressing this area. For example, He et al. [32] has proposed a roadside unit-satellite integrated vehicular network to provide vehicles with fresh environmental sensing data for safer driving. Wang et al. [33] has explored low earth orbit (LEO) satellite-assisted vehicular network, aiming at minimizing overall system delay through optimized task scheduling and offloading strategies. Additionally, an energy-efficient crowdsensing framework has been proposed that combines uncrewed ground vehicles (UGVs), uncrewed aerial vehicles (UAVs), and satellites for data collection, particularly in post-disaster scenarios [22]. Furthermore, a space-air-ground integrated VCS system with a blockchain-based incentive mechanism has been examined [12], where social welfare is maximized through vehicle winner selection. However, satellites and UAVs are introduced at the architectural level, and their participation is not explicitly modeled or optimized in the incentive design, making the incentive mechanism very generic and largely independent of the space-air layer.

Although a variety of incentive mechanisms have been proposed for VCS, most existing designs are tailored for specific system assumption or terrestrial settings. These mechanisms are therefore not directly applicable to the satellite-assisted VCS setting.

B. DRL for Vehicular Network

DRL has emerged as a powerful and widely adopted tool in vehicular network for addressing a variety of complex decision-making problems such as task and computation offloading [34], [35], [36], [37], [38], [39], [40], resource allocation [37], [38], [41], [42], [43], vehicle selection in vehicular edge computing (VEC) [44], [45], and network security [46]. For example, proximal policy optimization (PPO) has been proposed to intelligently schedule when and where vehicular computation tasks should be

executed, effectively balancing latency and energy consumption under dynamic wireless conditions and vehicle mobility [34]. Similarly, DRL has been employed to select road side units (RSUs) responsible for task reception, collaborative computation, and result delivery in VEC network [35]. Considering infrastructure limitations, a DRL-based computation offloading framework has been further proposed for UAV-assisted vehicular edge network [36]. While the aforementioned studies primarily focus on task or computation offloading decisions, subsequent works have extended DRL frameworks to jointly optimize offloading and resource management. In particular, asynchronous DRL has been adopted to jointly optimize collaborative task offloading and on-demand resource allocation across vehicles, edge servers, and cloud, aiming to minimize system-wide task execution delay under high mobility and resource heterogeneity [37]. Along similar lines, DRL has also been applied to joint task scheduling and resource allocation in dynamic VEC, where offloading decisions, priority assignment, and transmit power control are optimized to reduce task completion time and energy consumption [38].

Beyond computation-centric applications, intelligent vehicular systems generate heterogeneous services with conflicting quality of service (QoS) requirements. To address this challenge, network slicing in vehicular networks has been formulated as a dynamic resource allocation problem among heterogeneous slices, and DRL has been adopted to adaptively allocate resources under varying traffic conditions [41]. Moreover, DRL has been explored for joint relay selection and transmission power allocation in multi-hop millimeter-wave (mmWave) device-to-device (D2D) vehicular communications [42].

In addition to task and computation offloading, and resource allocation, DRL has also been explored for content caching [47], [48], [49] in VEC networks. For example, DRL has been employed to optimize cooperative content caching in VEC by minimizing long-term content transmission delay, while achieving mobility awareness and privacy preservation through federated learning [47]. Alongside performance-oriented applications, DRL has been investigated to enhance security in vehicular networks. Specifically, a DRL-based method has been proposed to maximize the sum secrecy rate in secure mmWave vehicular networks by exploiting idle vehicles as cooperative relays and jammers, thereby improving resilience against eavesdropping attacks [46].

Federated learning (FL) has recently gained significant attention in vehicular networks as an effective approach for enabling collaborative model training while preserving data privacy. However, the high mobility of vehicles often lead to frequent client dropouts and straggler effect, which can severely degrade FL training efficiency and model performance. To address these challenges, DRL has been increasingly adopted to enhance FL system operations. For example, a reliability-aware twin delayed deep deterministic policy gradient (TD3) scheme has been proposed to jointly optimize vehicle selection along with central processing unit (CPU) and uplink bandwidth allocation, thereby improving the robustness of FL training under vehicle volatility [44]. Similarly, a vehicle selection method based on deep Q-learning (DQN) has been proposed to maximize the global model accuracy while minimizing processing time and communication overhead in vehicular FL systems [45]. Beyond participation selection, DRL has also been leveraged to improve FL aggregation process itself [50]. In this study, a DRL based approach has been proposed to adaptively optimize aggregation weights for local model updates.

C. MADRL for Vehicular Network

MADRL has been extensively investigated in vehicular networks to address complex decision-making problems involving multiple autonomous agents, and has also been explored in UAV-assisted crowdsensing systems [51]. There are studies focused on task scheduling and computation offloading in edge-assisted vehicular environments [52], [53], [54], [55]. For instance, a CTDE-based MADRL framework has been proposed to enable cooperative, hierarchical task scheduling among heterogeneous UAVs, where each UAV agent is responsible for task reception and task retransmission, effectively minimizing latency and improving load balance in UAV-assisted VEC networks [52]. Along similar lines, vehicular computation offloading has been formulated as a MADRL problem in which vehicles autonomously select mobile edge computing (MEC) servers to minimize long-term task execution delay under dynamic mobility and resource contention [53]. Furthermore, a hybrid DRL framework has been proposed to jointly learn pricing and computation offloading strategies without requiring private information sharing, where edge server employs DRL to learn dynamic pricing policies, and vehicles act as independent agents using MADRL to determine how much computation to offload [54]. In a related direction, MADRL has been applied to vehicular fog computing (VFC), where neighboring regions share idle computing resources through RSUs acting as relays, and task vehicles learn distributed offloading policies to avoid spatio-temporal resource waste [55].

Several works have extended MADRL to jointly optimize offloading and resource allocation. For instance, a multiple-input multiple-output (MIMO)-non-orthogonal multiple access (NOMA) VEC system has been considered that allows multiple vehicles to offload tasks over the same spectrum. Due to vehicle mobility, inter-vehicular interference introduced by MIMO-NOMA, and stochastic tasks arrivals, the wireless channel conditions and system load become highly uncertain, making power allocation extremely difficult. To tackle this issue, a decentralized MADRL-based approach has been proposed, in which each vehicle independently learns its power allocation strategy for local computation and task offloading to minimize long-term power consumption and task latency [43]. Similarly, a NOMA-based VEC framework has been developed that combines adaptive distance and channel-aware resource allocation with MADRL, where distributed edge nodes independently learn cooperative offloading and resource allocation [56]. To support heterogeneous QoS requirements, MADRL has been proposed to jointly optimize channel allocation and power control for vehicle-to-vehicle (V2V)-communications, enabling delay-sensitive and delay-tolerant vehicular services [57]. To further improve scalability in dense vehicular environments, a

digital-twin-assisted MADRL framework has been proposed, in which vehicles are adaptively grouped into manageable structures to simplify the learning environment, enabling efficient task offloading and resource allocation in large-scale vehicular networks [58].

Beyond computation offloading and resource allocation, MADRL has been explored for content caching. A collaborative caching strategy has been proposed in which vehicles act as caching nodes and collaboratively learn optimal cache replacement policies using MADRL, thereby reducing content access delay and backhaul usage [59]. Moreover, MADRL has also been explored for security and safety applications in vehicular networks, such as collaborative sensor anomaly detection in autonomous driving system, enabling decentralized identification mitigation of malicious or faulty vehicles [60].

D. Asynchronous MADRL

In most of the above studies, DRL is employed for optimization and real-time decision-making [3], [15]. However, standard MADRL frameworks typically assume synchronous actions and uniform episode lengths across all agents [61], [62]. When agents finish their tasks early, this requires per-agent termination handling and passing inactive agents with zero rewards until all agents finish their task. Such design choices simplify joint trajectory collection and centralized-critic updates, but impose constraints that may not reflect real-world asynchronous behavior.

A few studies have explored asynchronous MADRL settings to better reflect realistic multi-agent behavior. One work investigated heterogeneous agents and introduced individual centralized critics to update each agent's policy independently [63] without waiting for other agents to finish their tasks. Liang et al. [61] has considered that agents interact with the environment at different times and complete tasks at varying rates. In this setup, trajectories of idle agents are not stored, and policy updates are triggered only when all agents' battery levels fall below a predefined threshold. A further study proposes an asynchronous MADRL framework that supports irregular trajectory collection by skipping idle time steps and storing data only when decisions occur [64]. As a result, agents accumulate different amounts of experience within a fixed 600-second window. To mitigate the resulting data imbalance, the authors adopt a CTDE approach enabling knowledge sharing so data-poor agents can benefit from data-rich ones. Additionally, another work examines a setting where agents interact with the environment simultaneously but update their policies sequentially in a round-robin manner, thereby avoiding the instability that can arise from simultaneous multi-agent updates [65].

It should be noted that in all of these reported works, the asynchronous MADRL designs differ from one another in terms of environment design and training strategy, as they are inherently problem specific.

III. System Model & Problem Formulation

In this paper, we define sensing level as volume of sensing data collected in the VCS game at each timestep. The key notations used in this paper are summarized in Table II. Let $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the set of vehicles. Each vehicle $i \in \mathcal{N}$ is equipped with numerous sensors and independently determines its sensing level x_i^t at the time step t to maximize its own utility. Let $\mathbf{x}^t = (x_1^t, x_2^t, \ldots, x_N^t)$ denote the vector of sensing levels of all vehicles and $\mathbf{x}_{-i}^t = \{x_1^t, x_2^t, \ldots, x_{i-1}^t, x_{i+1}^t, \ldots, x_N^t\}$ represents the sensing levels of all vehicles except vehicle i at time step t.

TABLE II
Key Notations

Notation	Definition
N	Number of vehicles
x_i^t	Sensing level of i-th vehicle
$\hat{r}_i$	Sensing task payoff
$h_p^i(t)$	Channel coefficient between the p-th antenna of the i-th vehicle and the satellite
f_c	Carrier frequency
$d_p^i(t)$	Distance between the p-th antenna of the i-th vehicle and the satellite
c	Velocity of light
τ	Path-loss exponent
$h_p^{LoS,i}(t)$, $h_p^{NLoS,i}(t)$	LoS, NLoS Channel component between the p-th antenna of the i-th vehicle and the satellite
ψ^{LoS}, ψ^{NLoS}	Parameters for LoS and NLoS propagation
β	Rician factor
R	Earth's radius
r	Orbital altitude of the satellite
$\omega(t)$	Geocentric angle
$\alpha(t)$	Elevation angle
λ	Wavelength
a	Spacing between antennas
θ_{AoA}	Angle of arrival
$\mathcal{CN}$	Complex Gaussian distribution
γ_i^t	SNR of i-thg vehicle
P_0	Total transmit power
B	System bandwidth
N_0	Noise power spectral density
$R_i(t)$	Achievable data rate
G	Social network graph
g_{ij}	Relation between vehicle i and vehicle j
Z, L	In-degree, out-degree vector respectively
g	Total number of edges in the network
W	Social network matrix
w_{ij}	Influence vehicle j has on vehicle i
u_i^t, η_i^t, f_i^t	Total, extrinsic, intrinsic reward respectively
q_i^t	Quality factor
c_i	Sensing unit cost
e_i^t	Remaining battery level
z_i^{in}	Inbound influence
z_i^{out}	Outbound influence

A VCS server is deployed on a LEO satellite serving a remote area where terrestrial service is not available, and vehicles are directly connected with the satellite-based crowdsensing server, as shown in Fig. 1. The satellite-based VCS server considers a set of sensing tasks, each associated with a payoff $\hat{r}_i$. Each vehicle is associated with exactly one sensing task, and the vector of task payoffs associated with vehicles is denoted by $\mathcal{R} = \{\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_N\}$.

A. Vehicle-Satellite Channel Model

We consider a vehicular network where each vehicle is equipped with P antennas and communicates with a LEO satellite equipped with a single antenna. It is assumed without loss of generality that during the entire sensing duration, all vehicles remain within the coverage region of the satellite on which the corresponding VCS server is mounted, and transmit their sensing data to the VCS server [66].

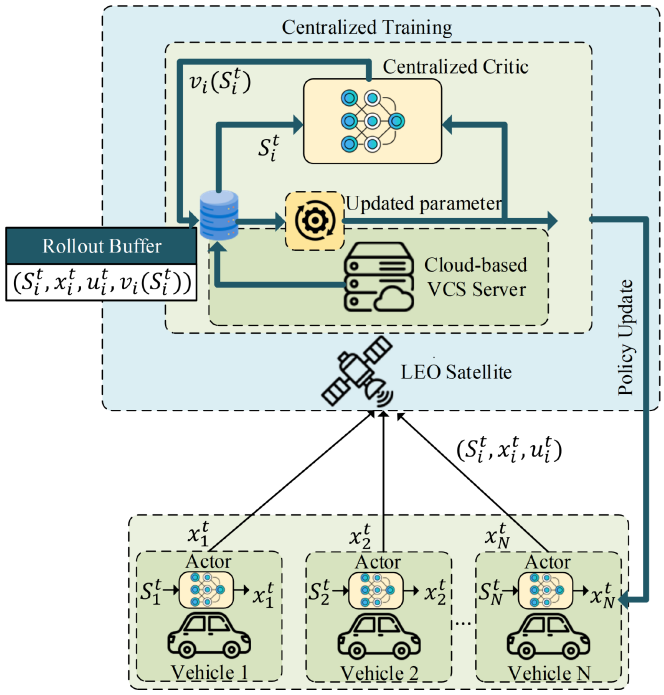

Fig. 1. System model of satellite-based vehicular crowdsensing.

The wireless link between the vehicles and the satellite is modeled as a ground-to-air channel, where the free-space path loss model is considered for large-scale path loss, and the Rician model is considered for small-scale path loss [67]. The channel coefficient between the p-th antenna of the i-th vehicle and the satellite at time t can be expressed as

$$h_p^i(t) = \left(\frac{4\pi f_c d_p^i(t)}{c}\right)^{-\frac{\tau}{2}} \cdot \left(\psi^{\text{LoS}} h_p^{\text{LoS},i}(t) + \psi^{\text{NLoS}} h_p^{\text{NLoS},i}(t)\right), \quad (1)$$

where c denotes the speed of light, f_c is the carrier frequency, and τ is the path-loss exponent. The distance between the p-th antenna of the i-th vehicle and the satellite at time t, denoted as $d_p^i(t)$, can be calculated as [68]

$$d_p^i(t) = \sqrt{R^2 + (R+r)^2 - 2R(R+r)\cos(\omega(t))}, \quad (2)$$

where R is the Earth's radius, r is the orbital altitude of the satellite, and $\omega(t)$ is the geocentric angle between the i-th vehicle and the satellite at time t given by [68]

$$\omega(t) = \cos^{-1}\left(\frac{R}{R+r}\cos(\alpha(t))\right) - \alpha(t), \quad (3)$$

where $\alpha(t)$ denotes the elevation angle between the i-th vehicle and the satellite at time t. The parameters for line-of-sight (LoS) propagation ψ^{LoS} and for non-line-of-sight (NLoS) propagation ψ^{NLoS} are defined as

$$\psi^{\text{LoS}} = \sqrt{\frac{\beta}{\beta+1}}, \qquad \psi^{\text{NLoS}} = \sqrt{\frac{1}{\beta+1}},$$

where β is the Rician factor. The terms $h_p^{\text{LoS},i}(t)$ and $h_p^{\text{NLoS},i}(t)$ represent the LoS and NLoS channel components, respectively [67], [69] as follows:

$$h_p^{\text{LoS},i}(t) = e^{\frac{-j2\pi}{\lambda}a(p-1)\cos(\theta_{AoA}(t))}, \quad (4)$$

$$h_p^{\text{NLoS},i}(t) \sim \mathcal{CN}(0,1), \quad (5)$$

where λ is the wavelength, a is the spacing between antennas, and $\theta_{AoA}(t)$ is the angle of arrival of a signal from vehicle to satellite at time t. Furthermore, in (5), the NLoS component follows a complex Gaussian distribution, with zero means and unit variance [70].

The instantaneous signal-to-noise ratio (SNR) between the i-th vehicle and the LEO satellite can be expressed as [71]

$$\gamma_i(t) = \frac{P_0 \left|\sum_{p=1}^{P} h_p^i(t)\right|^2}{BN_0}, \quad (6)$$

where P_0 denotes the total transmit power of the vehicle, B is the system bandwidth, and N_0 represents the single-sided noise spectral density.

Then, the achievable data rate of the link between i-th vehicle and the LEO satellite is given by [67]

$$R_i(t) = B\log_2(1+\gamma_i(t)). \quad (7)$$

B. Social Network Model

We consider that all vehicles virtually form a social network i.e., virtual graph, generated by the Erdős–Rényi (E–R) model [72], in which each pair of nodes is connected with probability μ. A larger value of μ means a denser network topology with increased interactions among vehicles, which enables more coordinated sensing decisions. Conversely, a lower μ leads to sparser connectivity and more independent sensing behavior. The social network is represented by an adjacency matrix $G = [g_{ij}]_{N\times N}$, where $g_{ij} \in \{0,1\}$ indicates whether vehicle j is connected with vehicle i, and self loop connections are excluded, i.e, $g_{i,i} = 0$.

There are two cases regarding whether the vehicles have knowledge of the network structure. In the complete information setting, every vehicle has full knowledge of the network structure, meaning that each vehicle knows which vehicle is connected with whom. Here, however, we address a more challenging incomplete information scenario, where vehicles are unaware of the network structure. Instead, each vehicle only knows the distribution of in-degrees and out-degrees of all vehicles in the network. Let $\mathbf{1}$ be an N-dimensional column vector of ones. The in-degree vector is then expressed as $Z = G\,\mathbf{1}$ and the out-degree vector as $L = G^T\,\mathbf{1}$. The total number of edges in the network is given by $g = \mathbf{1}^T G\,\mathbf{1}$. Therefore, the social network influence matrix W can be obtained as [3], [15]

$$W = \frac{Z\,L^T}{g}, \quad (8)$$

where the entry of the matrix W, w_{ij} represents the influence vehicle j has on vehicle i.

Unlike [3], where the social network matrix is considered as static throughout the entire sensing period, we consider a time-varying network to introduce stochasticity into the environment and account for mobility-induced uncertainty. Once the social influence matrix W is initialized at the beginning of each sensing period, each element of the matrix $w_{ij}(t)$ evolves according to a Bernoulli-perturbation, with probability $\mathcal{P}$ [73], serving as a stochastic abstraction rather than an exact representation of real-world social evolution. The value is updated by a bounded random offset drawn from $\mathcal{U}(-b, b)$ and clamped to [0,1]. Otherwise, it remains unchanged.

$$w_{ij}(t+1) = \begin{cases} \text{clip}(w_{ij}(t) + y_{ij}(t),\ 0,\ 1)\,, \\ \qquad \text{with probability } \mathcal{P}, \\ w_{ij}(t), \text{with probability } 1-\mathcal{P}. \end{cases} \tag{9}$$

where the function $\text{clip}(p, q, r)$ limits its argument p to the interval $[q, r]$, and $y_{ij}(t) \sim \mathcal{U}(-b, b)$ is a uniformly distributed random variable bounded by $[-b, b]$.

C. Reward Model

In sensing model, each vehicle receives two types of rewards: an extrinsic reward η_i^t and intrinsic reward f_i^t. The extrinsic reward accounts for the direct benefits and costs associated with the sensing process, and the communication process. The intrinsic reward represents reward from social network effect, which captures the influence of neighboring vehicles. Specially, the intrinsic reward increases with the sensing levels of neighboring vehicles, meaning that a higher collective sensing effort among neighbors provides a higher incentive for vehicle i. Therefore, overall reward of vehicle i in the t-th time slot is expressed as

$$u_i^t(x_i^t, \mathbf{x}_{-i}^t) = \eta_i^t(x_i^t) + f_i^t(x_i^t, \mathbf{x}_{-i}^t), \tag{10}$$

where x_i^t is the sensing level of i-th vehicle. The term $\eta_i^t(x_i^t)$ represents the extrinsic reward received by vehicle i, which can be expressed as

$$\eta_i^t\left(x_i^t\right) = q_i^t x_i^t \hat{r}_i - c_i^t x_i^t - P_0 \frac{x_i^t}{R_i(t)}. \tag{11}$$

This first term of extrinsic reward is immediate task reward which is determined by the vehicle's sensing level x_i^t, the associated task payoff $\hat{r}_i$, and a quality score q_i^t. The quality score captures the usefulness of the collected sensing data, as not all sensed information is valuable; some may be noisy or redundant and therefore disregarded. In practice, the quality score can depend on several factors, such as the accuracy of the sensing data or reputation of the participating vehicle [74]. However, in this study, the design of the quality score is beyond the scope of the study. We adopt the same approach as [3] to obtain the quality score from previous sensing campaign.

The second and third terms in (11) represent the sensing cost and communication cost, respectively. The sensing cost is modeled as the product of the sensing amount x_i^t and the battery cost per unit of sensing c_i^t. So, the remaining battery budget for that task of the vehicle i decreases to

$$e_i^{t+1} = e_i^t - c_i^t x_i^t. \tag{12}$$

It is important to note that, the vehicle stops sensing once the battery level falls below a predefined threshold. The communication cost is defined as the product of transmission power P_0 and the transmission duration. where it is expressed as $x_i^t / R_i(t)$, i.e., the ratio between the sensed data size and the achievable data rate at time t [67].

The second term $f_i^t(x_i^t, \mathbf{x}_{-i}^t)$ in (10) is the intrinsic reward that comes from social effect and can be expressed as

$$f_i^t(x_i^t, \mathbf{x}_{-i}^t) = \sum_{j=1}^{N} w_{ij}(t) x_i^t x_j^t, \tag{13}$$

where the intrinsic reward gained by vehicle i from vehicle j is given by $w_{ij}(t) x_i^t x_j^t$ which is widely adopted in the social network research [75], [76], [77].

D. Problem Formulation

The sensing level decision-making process in the t-th time slot is modeled as a non-cooperative game. In this setting, each vehicle receives an extrinsic reward from its own sensing actions and an intrinsic reward derived from the sensing efforts of neighboring vehicles, and aims to maximize its cumulative reward over time.

In the t-th time slot, the vehicular sensing game is defined as the tuple

$$G^t = \{\mathcal{N}, \mathbf{x}^t, U^t\}, \tag{14}$$

where

- $\mathbf{x}^t$ represents the sensing profile of all vehicles, where the sensing amount of vehicle i is constrained within the range $[x_{i,\min}^t,\ x_{i,\max}^t]$. However, the sensing amount is not only constrained by the minimum and maximum sensing bounds but also by the available battery budget for that task. That is

$$x_i^t \in \mathbf{x}_i^t = \left[x_{i,\min},\ \min\left\{x_{i,\max}, f(e_i^t)\right\}\right], \tag{15}$$

  where $f(e_i^t)$ ensures that sensing stops when the battery drops below a threshold. The function $f(e_i^t)$ represents the maximum sensing level allowed by the available battery energy and is defined as

$$f(e_i^t) = \frac{e_i^t}{c_i^t + P_0 / R_i(t)}. \tag{16}$$

- $U^t = \{u_1^t, u_2^t, \ldots, u_N^t\}$ is the set of reward functions of all vehicles in the t-th time slot.

Given $\mathbf{x}^t$, the best response of the vehicle i in the t-th time slot is to choose the sensing level x_i^t such that its own utility is maximized. Mathematically, this can be expressed as

$$x_i^{t*} = \arg \max_{x_i^t \in \left[x_{i,\min},\ \min\{x_{i,\max}, f(e_i^t)\}\right]} u_i^t\left(x_i^t, \mathbf{x}_{-i}^t\right). \tag{17}$$

IV. Proposed Solution: Asynchronous CTDE MAPPO

A. State Space

For a given time step t, the state of the i-th vehicle can be represented as

$$S_i^t = \left\{ x_i^{t-1},\ z_i^{\text{out},t},\ z_i^{\text{in},t},\ \hat{r}_i, c_i^t, R_i(t),\ e_i^t \right\}, \tag{18}$$

where $z_i^{\text{in},t} = \frac{1}{N-1}\sum_{j\neq i} w_{ij}(t)$ and $z_i^{\text{out},t} = \frac{1}{N-1}\sum_{j\neq i} w_{ji}(t)$ denote the average inbound and outbound influences, respectively [78]. Here, inbound influence measures how much vehicle i is affected by others while outbound influence means how much vehicle i affects its neighbors.

B. Action Space

The action space for vehicle i at time t, denoted as $\mathcal{A}_i$ contains a single continuous decision variable representing optimized sensing level x_i^t where $x_i^t \in \mathcal{A}_i$. Formally, the action space is defined as

$$\mathcal{A}_i = \left[x_{i,\min},\ \min\left\{ x_{i,\max}, f(e_i^t) \right\} \right]. \tag{19}$$

C. Reward Function

In this study, the reward function shown in (10) is designed to quantify the reward u_i^t obtained by the agent for taking action x_i^t at state S_i^t. The rationale of the designed reward function is to encourage the agent to maximize the cumulative rewards over time, promoting optimal decision-making in vehicle sensing behaviors.

D. Asynchronous MAPPO in CTDE Manner

The proposed MAPPO is an extended version of the PPO algorithm tailored for MADRL scenarios. MAPPO follows an on-policy learning scheme that learns from the most recent experiences generated by the current policy [79]. Our proposed MAPPO algorithm operates under CTDE framework, where agents are trained collaboratively through centralized information sharing, while execute actions individually based on local observations. In CTDE framework, all agents are trained together which enables a central critic or centralized learning process to stabilize and speed up learning. In decentralized execution, each agent, once deployed, acts independently based solely on its local observation or partial view of the environment. Agents do not require access to others' actions or global state at runtime. Since all agents are homogeneous which means they all perform similar kind of tasks, parameter sharing is introduced to improve the convergence performance. The action space is continuous where a Beta distribution is used as the output of the policy network to fit the selection of actions where mean and variance can be trained through the neural network.

1) A-MAPPO Algorithm: In the A-MAPPO algorithm under the CTDE framework, there are three main components: the actor network π_{θ_i}, the critic network V_{ϕ_i}, and a shared rollout buffer $\mathcal{D}$. Within the CTDE paradigm, all homogeneous agents share the same set of parameters to facilitate coordinated learning. The diagram of the proposed A-MAPPO is shown in Fig. 2.

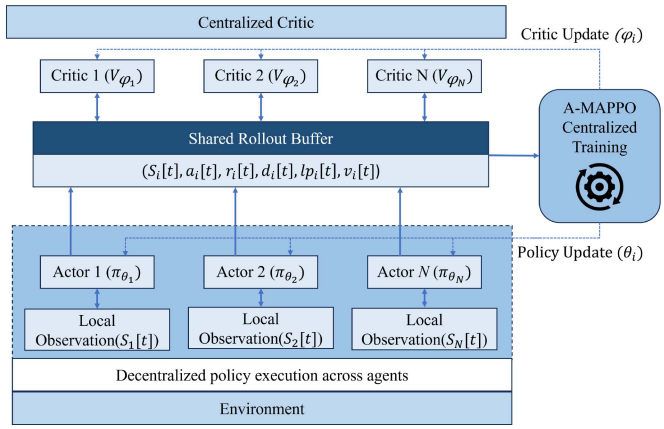

Fig. 2. A diagram of proposed A-MAPPO method.

To avoid confusion, we emphasize that the A-MAPPO algorithm and the system model use different notation, as detailed below. In particular, the algorithm description follows the standard DRL notation. Let θ_i denote the parameters of the actor network π_{θ_i} for agent $i \in \mathcal{I}$, representing the policy, and let ϕ_i denote the parameters of the critic network V_{ϕ_i}, which approximates the state-value function. Note that, since the agents are homogeneous and with centralized training, their parameters are identical across all agents, i.e., $\theta_i = \theta_j$ and $\phi_i = \phi_j, \forall i, j \in \mathcal{I}$.

The rollout buffer $\mathcal{D}$ has finite capacity and stores experience tuples of the form $\{S_i[t],\ a_i[t],\ r_i[t],\ d_i[t],\ lp_i[t],\ v_i[t]\},\ \forall i \in \mathcal{I}$, where $S_i[t]$ denotes observation, $a_i[t]$ is the action taken, $r_i[t]$ represents the reward received, $d_i[t]$ is the termination flag (done), $lp_i[t]$ is the log-probability associated with the sampling action $a_i[t]$, and $v_i[t]$ is the value predicted by the critic network V_{ϕ_i} in the timestep t.

This experience collection continues until the buffer reaches its maximum capacity. Following that, the experiences of all agents are first flattened and sampled in multiple mini-batches. Throughout this process, individual agents may complete multiple episodes before the buffer becomes full. Once the buffer is filled, A-MAPPO training is initiated, using experiences generated by the current policy, as required in on-policy reinforcement learning.

Before flattening and sampling experiences into mini-batches, the accumulated discounted return is first computed as

$$\delta_i[t] = r_i[t] + \gamma\, v_i[t+1] - v_i[t], \tag{20}$$

where $\gamma \in [0, 1]$ denotes the discount factor that controls the importance of future rewards. When $\gamma = 0$, the agent focuses solely on the immediate reward, disregarding long-term returns. When $\gamma = 1$, the agent considers the cumulative reward from the current time step t to the end of the episode.

Following that, the advantage is calculated, which represents how much better or worse an action is. To improve training stability, the generalized Advantage Estimator (GAE) method is employed to compute a smoothed estimate of the advantage function. Given the accumulated discount return $\delta_i[t]$ defined in

(20), the advantage for agent i at time step t is computed as

$$A_i[t] = \delta_i[t] + (\gamma\lambda)\,\delta_i[t+1] + \ldots + (\gamma\lambda)^{T-t-1}\delta_i[T-1], \tag{21}$$

where $\lambda \in [0,1]$ is the GAE parameter controlling the bias-variance trade-off, and T is the episode length.

The resulting advantage estimates are then normalized to improve stability as follows:

$$\hat{A}_i[t] = \frac{A_i[t] - \mu_A}{\sigma_A + \epsilon}, \tag{22}$$

where μ_A and σ_A denote the mean and standard deviation of the advantage values of agent i computed over the samples in the training batch, and $\epsilon = 1 \times 10^{-8}$ is a small constant added to prevent division by zero.

The normalized advantage $\hat{A}_i[t]$ is subsequently used to compute the policy gradient loss, allowing stable and efficient updates in the A-MAPPO framework.

- *Actor (Policy Network) Update:* A-MAPPO employs a clipped surrogate objective function to update the actor network to improve the policy so that it selects actions that return higher expected returns. It updates its actor network by maximizing a clipped surrogate objective, which is based on the ratio of the new and old policies, multiplied by the estimated advantage. The objective function is formulated as

$$J_i = \frac{1}{M}\sum_{m=1}^{M} \min\Big(\rho_i^m \hat{A}_i^m, \; \text{clip}(\rho_i^m,\, 1-\epsilon_A,\, 1+\epsilon_A)\,\hat{A}_i^m\Big), \tag{23}$$

where M represents the mini batch size, and m is the index of mini-batch. ϵ_A is a hyperparameter which limits the updating range. This clipping mechanism is central to A-MAPPO, where it is used to prevent excessive changes to the policy during training, ensuring that the updated policy does not deviate significantly from the previous one. The term ρ_i^m is the ratio of the new policy to the old policy from previous update can be represented as

$$\rho_i^m = \frac{\pi_{\theta_i}(a_i^m|s_i^m)}{\pi_{\theta_i}^{\text{old}}(a_i^m|s_i^m)}. \tag{24}$$

$$H(\pi_{\theta_i}) = \mathbb{E}\left[-\log\left(\pi_{\theta_i}(a_i \mid s_i)\right)\right]. \tag{25}$$

Hence, the objective function of actor network can be expressed as

$$J_i^A = J_i + c_{\text{ent}} H(\pi_{\theta_i}), \tag{26}$$

where c_{ent} is the entropy coefficient.

This ratio shows how much more or less likely the new policy is to choose the same action compared to the old policy. By incorporating this ratio into the clipped surrogate objective, A-MAPPO maintains a balance between policy improvement and training stability. In order to encourage exploration of the policy, a policy entropy is introduced with the actor loss. First, the policy entropy is multiplied by an entropy coefficient and then added to the actor network's loss. The policy entropy is calculated as follows:

- *Critic Update:* The critic loss makes the value function predict the discounted return accurately. The critic loss can be expressed as follows:

$$L_{V_{\phi_i}} = \frac{1}{M}\sum_{m=1}^{M}\left(v_i^m - \delta_i^m\right)^2. \tag{27}$$

After that, the actor and critic losses are combined into a single total loss, which is backpropagated once to ensure simplicity and efficiency. The overall objective is to maximize the actor's objective function while minimizing the critic's loss function.

2) Algorithm Description: The A-MAPPO-based sensing level optimization procedure for VCS is summarized in Algorithm 1. In this approach, experiences $\{s_i[t],\, a_i[t],\, r_i[t],\, d_i[t],\, lp_i[t],\, v_i[t]\},\ \forall i \in \mathcal{I}$ for all agents are continually collected until the rollout buffer reaches its full capacity, where the capacity of the buffer, $\mathcal{D}$ defines the number of exploration steps per collection period. This buffer may contain experience spanning multiple episodes for each agent, meaning that agent updates do not occur at the end of every episode but when the buffer is full.

The environment is designed for asynchronous multi-agent dynamics where each agent operates independently. So, individual agent may finish their episode at different times. When one agent finishes an episode before the others, the environment of only that agent is reset and its episode counter is incremented. The remaining agents continue their current episodes uninterrupted. As a result, upon buffer completion, the rollout buffer contains equal number of experiences per agent, but these samples will generally correspond to different episodes for each agent. This mechanism ensures efficient experience utilization and proper support for asynchronous episode boundaries during MAPPO training.

In the implementation, the done flags are carefully handled to mark episode boundaries for each agent. This careful handling ensures that return and advantage calculations are accurate, even when episodes end asynchronously across agents. Properly tracking the done flags is crucial for correctly resetting episode-specific statistics and for bootstrapping value estimation at episode ends.

After the rollout buffer is full, the returns and advantages are calculated for all experiences. However, prior to return δ_i and advantage A_i^t calculation, rewards r_i are normalized to stabilize training as follows:

$$r_i = \frac{r_i - \mu_{r_i}}{\sigma_{r_i} + \epsilon}. \tag{28}$$

Reward normalization plays a critical role in stabilizing and accelerating the training process. This normalization ensures that rewards have zero mean and unit variance, which mitigates issues caused by large variations in reward scales and stabilizes gradient updates.

Following that, all the experiences collected are flattened first across all agents and sampled into mini-batches, and then the

A-MAPPO update begins. For each mini batch, the actor loss J_i and value loss $L_{V_{\phi_i}}$ are calculated and then added. Then the total loss is backpropagated to update the network parameters.

The same batch of experiences is used for multiple epochs of updates, allowing the policy and value networks to fully exploit the collected data. After completing all update epochs for all batches, the rollout buffer is cleared, and new experiences are collected to repeat this process.

E. Computation Complexity

Let K denote the number of update epochs. In each training iteration, the algorithm processed $N \times \mathcal{D}$ collected transitions. Therefore, the training complexity of the proposed A-MAPPO algorithm is approximately $\mathcal{O}(KN\mathcal{D}(C_\pi + C_V))$, where C_π and C_V denote the computational costs of the actor and critic networks, respectively. Since the critic is centralized, its cost grows with the number of agents. During execution, only decentralized actor is used, yielding an inference complexity of $\mathcal{O}(NC_\pi)$ for all agents.

V. Simulation Results and Discussions

In this section, numerical results are presented to evaluate the performance of the proposed asynchronous MAPPO algorithm under the CTDE framework for satellite-assisted dynamic VCS. First, the convergence behavior of the proposed model is shown under various implementation configurations, including the use or omission of reward normalization, and entropy coefficient decay. Subsequently, the convergence performance is compared against multiple baseline models, such as Random, DQN, Greedy-Q policy, followed by comparative analyses under different system parameters, such as the number of vehicles, task payoff, and transmit power.

A. Simulation & Training Settings

For simulation, we consider a system model comprising $N = 10$ vehicles, each equipped with $P = 8$ antennas and a transmission power of $P_0 = 30$ dBm or 1 W. The system includes a VCS server located on a LEO satellite positioned at a distance of $r = 500$ km from the Earth's surface and equipped with a single antenna.

Communication is established using a carrier frequency of $f_c = 2$ GHz, system bandwidth $B = 5$ MHz, and noise power $N_o = -174$ dBm/Hz. Channel propagation is modeled as Rician fading with a Rician factor $\beta = 5$, a pathloss exponent of $\tau = 2$, and an initial elevation angle α of 45 degrees. The task payoff is uniformly sampled from the range $[5, 10]$. All the simulation parameters are summarized in Table III and all the parameters associated with training the A-MAPPO algorithm under the CTDE manner are shown in Table IV.

All simulations are implemented in Python 3.12.9 using the PyTorch framework on a MacBook Pro equipped with an Apple M3 Pro processor and 18 GB of unified memory.

In this asynchronous setting, the training process differs from the conventional reinforcement learning, where agents typically train for a fixed number of episodes (e.g., 1000). Instead, we adopt an iterative training approach in which agents are trained over 250 iterations, with each iteration encompassing several episodes. An episode is terminated when either the maximum episode length of 60 time steps is reached or the battery budget is exhausted. Each vehicle is assigned a battery budget of 100 units per task. One iteration of training is considered complete when the rollout buffer reaches its full capacity, corresponding to 256 environment steps. Since agents operate asynchronously, different agents may complete a varying number of episodes by the end of training. To ensure consistency in evaluation, we consider only the first 1000 completed episodes for each agent in the analysis.

TABLE III
System Parameters [33], [67]

Parameter	Value
Number of vehicles, N	10
Number of antennas at vehicle, P	8
Number of antennas at satellite	1
Earth radius, R	6371 km
Distance to LEO satellites orbit, r	500 km
Transmission power of each vehicle, P_0	1 W
System bandwidth, B	5 MHz
Noise power, N_o	-174 dBm/Hz
Carrier frequency, f_c	2 GHz
Rician factor, β	5
Elevation angle, α	45
Path loss exponent, τ	2
Task payoff, $\hat{r_i}$	[5, 10]
Quality factor, q_i^t	[5, 10]

TABLE IV
A-MAPPO Training Parameters

Parameter	Value
Learning rate, l	0.0003
Number of hidden layers	2
Nodes per hidden layer	128
Activation function	ReLU, Beta
Optimizer	Adam
Rollout buffer size, $\mathcal{D}$	256
Mini-batch size, M	64
Epochs, K	16
Discount factor, γ	0.99
GAE smoothing parameter, λ	0.95
Entropy coefficient, c_{ent}	0.01
Clipping hyperparameter, ϵ_A	0.2

B. Baseline Models

Following [3], we adopt DQN, Greedy-Q, and Random approaches as baselines for fair comparison. The incentive mechanisms and asynchronous models reviewed in II-A and II-D are not included as baselines, as they are designed for problem settings and system assumptions that differ fundamentally from ours.

1) Random Approach: In the Random approach, each vehicle independently selects its sensing level from a predefined sensing range. If a vehicle's battery level drops below zero, it ceases sensing activity while others continue until they also run out of battery or reach the predefined episode length of 60 time steps.

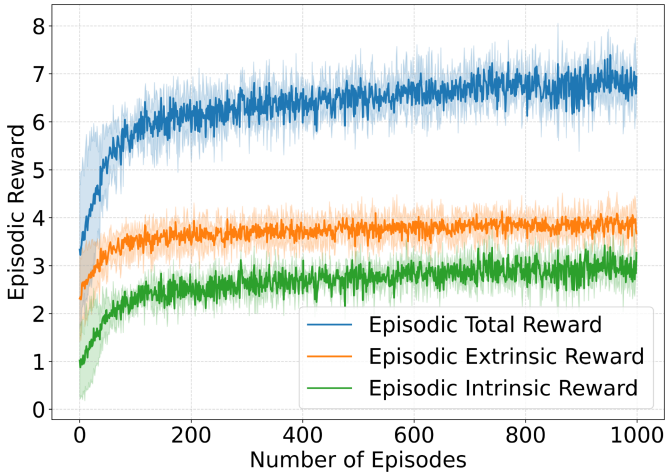

Fig. 3. Empirical performance of proposed A-MAPPO algorithm.

2) DQN: DQN is commonly adopted for decision-making tasks in vehicular sensing [80] and vehicular crowdsensing [53], [81] applications. While DQN is naturally designed for discrete action spaces, we adapted DQN to a continuous action setting by force discretization of the available sensing levels. This modification enables vehicles to make context-aware sensing decisions.

3) Greedy Q-Policy: As each vehicle's reward computation depends on the actions taken by other vehicles in the environment, traditional greedy algorithms are not directly applicable in our scenario. To address this, we adopted a Greedy Q-policy by stopping exploration within the DQN framework. In this method, the agent selects the action with the highest Q-value at each decision step, based on current knowledge, thereby ignoring future reward and maximizing immediate expected reward.

4) Soft Actor-Critic (SAC): We employed a multi-agent extension of Soft actor-critic (SAC) as an additional baseline to evaluate the performance of the proposed approach. SAC is an off-policy DRL algorithm that jointly optimizes expected reward and policy entropy, enabling improved exploration and sample efficiency in continuous action spaces.

C. Numerical Results and Performance Evaluation

1) Empirical Performance:

- *A-MAPPO performance:* As outlined in III-C, the total reward received by each vehicle is composed of an extrinsic reward and an intrinsic reward.
  Fig. 3 shows the average episodic total, intrinsic, and extrinsic rewards of all vehicles. Both intrinsic and extrinsic rewards increase steadily throughout training, with a sharp rise in the first 200 episodes, reaching approximately 2.5 for intrinsic and 3.7 for extrinsic reward by episode 200. After this point, their growth becomes more gradual. Consequently, the total reward, which reflects the sum of intrinsic and extrinsic components, increases rapidly to just over 6 by episode 200 and then continues to increase slowly, following the same trend. By the end of 1000 episodes, the total reward reaches to nearly 7, indicating stable learning performance.
  To ensure stable policy updates despite variable episode lengths, training is triggered only when the shared rollout buffer reaches its full capacity, thereby guaranteeing a consistent number of experience samples across agents in each update. The collected transitions, although originating from different episodes and agents, are treated uniformly during mini-batch updates, which aligns with the on-policy learning requirement of PPO. Furthermore, the CTDE framework enables knowledge sharing across agents through a shared critic, which further stabilizes the learning process under asynchronous conditions.
- *Reward Normalization:* From the Fig. 4(a), it can be seen that when the reward normalization is adopted, it substantially improves the convergence behavior of the A-MAPPO algorithm. With reward normalization, A-MAPPO achieves rapid and almost stable convergence within approximately 200 episodes. In contrast, without reward normalization, A-MAPPO exhibits an oscillatory behavior across entire training period and fails to stabilize. Furthermore, reward-normalized A-MAPPO achieves higher episodic rewards, reaching values close to 7, whereas the non-normalized variant rarely surpasses 6.5 after 1000 episodes. These results highlight the critical role of reward normalization in enhancing both stability and overall performance. As reward is scaled to have zero mean and unit variance prior to updates, it does not allow the model to have excessively large gradient updates. This reduces variance, and provides a more consistent optimization. As a result, the learning becomes more robust, allowing for faster and more consistent policy improvement during training.
- *Varying Entropy Coefficient:* The entropy coefficient c_{ent} in A-MAPPO controls how much the agent explores different actions by adding randomness to its decisions. By keeping the agent's actions less predictable, it helps the learning process avoid getting stuck with limited or suboptimal behavior and instead encourages discovering better strategies before settling on the most rewarding ones. As shown in Fig. 4(b), both A-MAPPO with a varying entropy coefficient and A-MAPPO with a constant entropy coefficient exhibit similar convergence rates, achieving almost stable performance after approximately 200 episodes. However, the use of a varying entropy coefficient enables A-MAPPO to achieve higher episodic rewards compared to a constant entropy coefficient setting. By episode 1000, A-MAPPO using variable entropy coefficient stabilizes at an episodic reward close to 7, whereas the constant entropy coefficient A-MAPPO stops near 6, a difference of roughly 1.0. This is because dynamically adjusting the entropy coefficient effectively balances exploration and exploitation throughout training, helps agents to discover policies that returns higher rewards.

2) Convergence Performance Comparison: Fig. 5 presents a comparative analysis of episodic reward convergence for

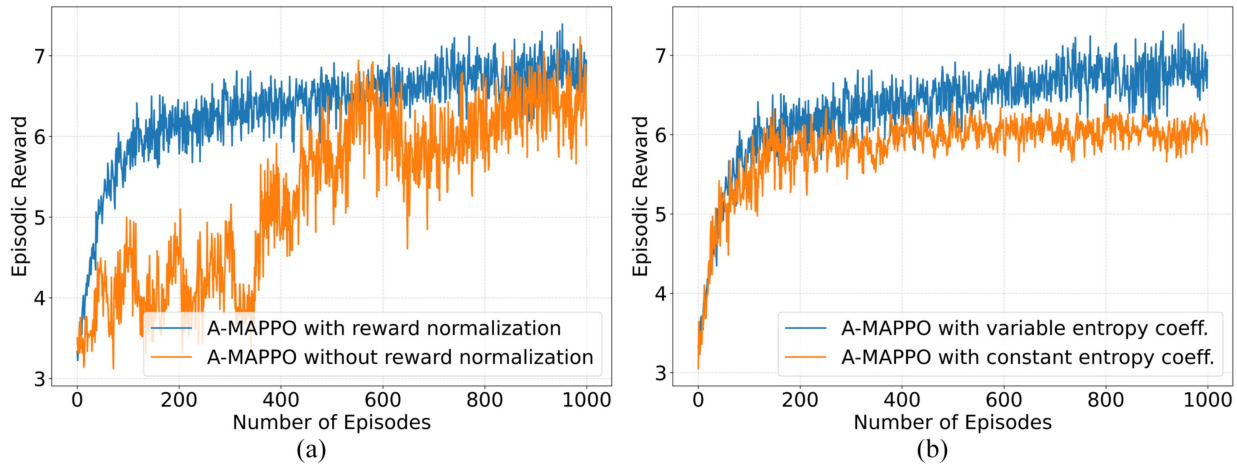

Fig. 4. (a) Convergence performance comparison for reward normalization vs no reward normalization, (b) Convergence performance comparison varying and constant entropy coefficient.

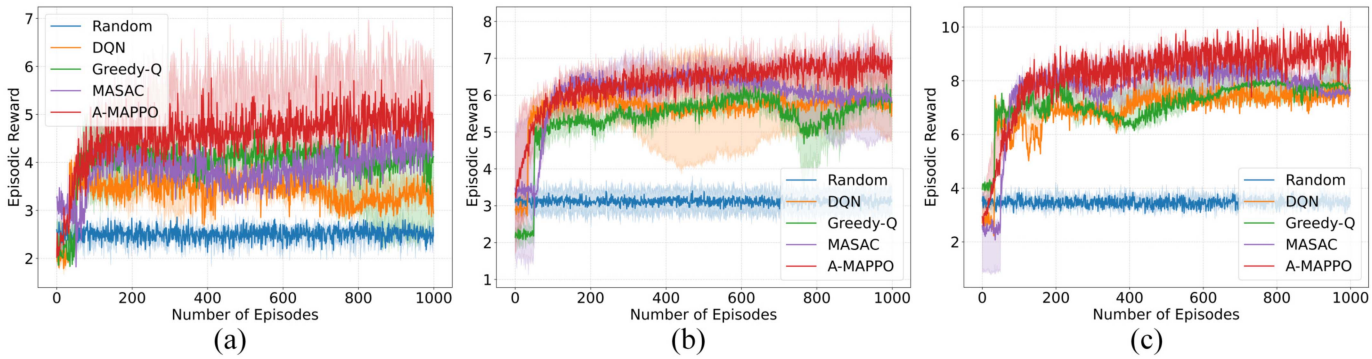

Fig. 5. Convergence performance comparison of different models for (a) $N = 5$, (b) $N = 10$, (c) $N = 15$ vehicles.

the proposed A-MAPPO algorithm, as well as MASAC, DQN, Greedy-Q, and Random policy baselines, under a social network probability of $\mu = 0.7$ and dynamic task payoff randomly sampled from [5, 10]. The results are further differentiated by the number of vehicles: (a) $N = 5$, (b) $N = 10$, and (c) $N = 15$.

Across all cases, the proposed A-MAPPO algorithm consistently demonstrates superior convergence behavior and achieves better episodic reward compared to the baseline models. For each vehicle count, A-MAPPO achieves almost stable performance in fewer than 200 episodes and maintains a significant reward gain over DQN, Greedy-Q policy, and random approaches. While DQN, and Greedy-Q also exhibit improved stability after 200 episodes, their achieved rewards are notably lower than A-MAPPO. Among the baselines, MASAC demonstrates competitive performance; however, its learning behavior exhibits higher variability and does not consistently match the performance of A-MAPPO.

For the smallest network ($N = 5$), DQN stabilizes relatively quickly but converges to a lower reward. Greedy-Q policy converges within 200 episodes, but its learning process becomes unstable after approximately 800 episodes, leading to performance degradation. MASAC demonstrates performance comparable to the Greedy-Q policy; however, it experiences a noticeable decline between 400-600 episodes, indicating reduced stability in middle stages of training. In contrast, A-MAPPO exhibits more stable learning behavior and yields higher average reward. As the number of vehicles increases to $N = 10$ and $N = 15$, A-MAPPO's advantage becomes more noticeable, maintaining a clear and robust margin over all competitors except MASAC. In these scenarios, Greedy-Q and DQN exhibit comparable average rewards, yet both remain substantially below A-MAPPO's performance gain. The Random policy, as expected, lags well behind all other approaches and fails to demonstrate meaningful convergence or reward improvement. For $N = 10$, A-MAPPO rapidly rises to an average of 6 by episode 200 and stabilizes just below 7.0 by episode 1000. MASAC achieves competitive performance in this setting, although its final performance remains slightly below that of A-MAPPO, converging to approximately 6.0. In comparison, Greedy-Q policy and DQN converge to values below 6.0, while Random at roughly 3.2. For $N = 15$, A-MAPPO reaches approximately 9.0 by episode 1000 whereas Greedy-Q reaches approximately 7.8, DQN stabilizes near 7.6, and Random ends around 3.7. At the early stages of training, MASAC exhibits similar performance as of A-MAPPO; however, after approximately 800 episodes, its performance declines and stabilizes around 7.7. A-MAPPO achieves approximately 28.6% higher average reward than Greedy-Q and about 16.9% improvement over DQN.

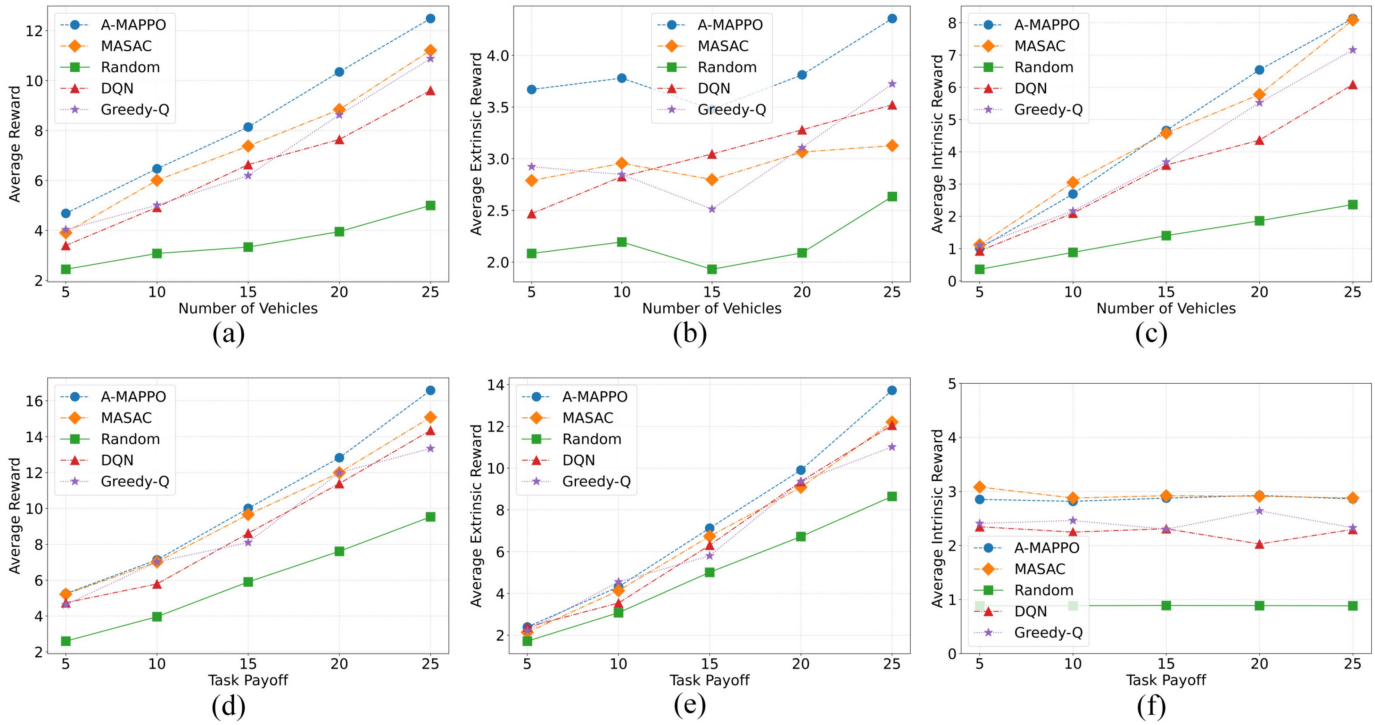

Fig. 6. Average total rewards, extrinsic rewards, and intrinsic rewards varying (a)–(c) number of vehicles, and (d)–(f) task payoff.

3) Impact of Number of Vehicles: Fig. 6(a)–(c) illustrates the impact of increasing the number of vehicles on the average reward and its extrinsic and intrinsic components. As the fleet size grows from $N = 5$ to $N = 25$, all approaches exhibit almost a linear upward trend in total rewards. Among all methods, A-MAPPO constantly achieves the higher reward across all network sizes, even when N increases as shown in Fig. 6(a). For example: when $N = 5$, A-MAPPO attains an average reward of approximately 4.8, which is about 20% higher than the second best performing Greedy-Q baseline (reward ≈ 4.0). As the network size increases to $N = 25$, A-MAPPO achieves an average reward of approximately 12.5, outperforming Greedy-Q policy (reward ≈ 10.8) by nearly 16%, and MASAC (reward ≈ 11.2) by nearly 10%. This highlights A-MAPPO's superior scalability and ability to effectively coordinate a growing number of vehicles. Random approach as expected lags behind all of the models by large margin.

The increasing trend in the average reward is primarily driven by the intrinsic reward component, as shown in Fig. 6(c). According to (13), larger vehicular populations naturally yield greater intrinsic benefits, resulting in approximately linear increase of the intrinsic reward with respect to N, as observed in the simulation results. For instance, the average intrinsic reward of A-MAPPO increases from approximately 1.0 at $N = 5$ to about 4.7 at $N = 15$ and further rises to nearly 8.2 when $N = 25$.

The average extrinsic reward however exhibits a non-monotonic trend as the number of vehicle increases, as shown in Fig. 6(b). As the fleet size grows from $N = 10$ to $N = 15$, a temporary reduction in the average extrinsic reward can be seen as vehicles are receiving incentives from intrinsic reward except DQN. For A-MAPPO, the extrinsic reward decreases from approximately 3.78 at $N = 10$ to about 3.5 at $N = 15$. However, when the number of vehicles is further increased to $N = 25$, improved learning allows the extrinsic reward to recover to approximately 4.3. Similar trends are observed for MASAC, DQN and Greedy-Q policy at lower reward levels, indicating the extrinsic reward is more sensitive to number of vehicles than intrinsic reward.

4) Impact of Task Payoff: Fig. 6(d)–(f) represents the average total, extrinsic and intrinsic rewards varying the task payoff when the vehicle number is $N = 10$, where all tasks are assigned the same payoff. As the task payoff increases from 5 to 25, all methods exhibit almost a linear improvement in average reward, driven by average extrinsic reward as shown in Fig. 6(d)–(e). Among all approaches, A-MAPPO consistently achieves the highest average total reward and the highest average extrinsic reward, while also demonstrating the steepest growth rate. Specially, A-MAPPO's average reward increases from approximately 5.2 at a payoff of 5 to about 16.7 at a payoff of 25. MASAC also shows strong performance, increasing from around 5.2 to approximately 15.0 over the same range, remaining competitive but consistently below A-MAPPO. In comparison, DQN improves from roughly 4.6 to 14.3, while Greedy-Q rises from 4.7 to 13.4 over the same range. The random policy remains substantially lower, increasing only from 2.6 to 9.5. At the highest task payoff, A-MAPPO achieves approximately 11% higher average reward than MASAC, around 17% higher than DQN and about 25% higher reward than Greedy-Q policy, highlighting its superior ability to learn though coordinated decision making.

Algorithm 1: A-MAPPO Training Algorithm for Satellite-assisted Vehicular Crowdsensing.

Initialize: Actor (π_θ) and critic model (V_ϕ), shared rollout buffer ($\mathcal{D}$), optimizer
Set episode, step counters and reward accumulators to zero
for each training iteration **do**
  Reset environment, obtain initial states for all vehicles
  for step $(t) = 1$ to $\mathcal{D}$ **do**
    For each agent, use current policy to sample actions: $a \leftarrow \pi_\theta(s)$
    Step the environment using action a to obtain s', r in (10), and *dones*.
    Insert $(s, a, r, dones, lp, v)$ for all agents into the shared buffer
    Accumulate per-agent episode rewards
    Reset logic: (Episode termination)
    for each agent i **do**
      if (episode length reaches 60) or (battery budget is exhausted) **then**
        $dones_i \leftarrow$ **True**
      end if
    end for
    if any agent is done **then**
      if all agents done **then**
        Increment episodes, reset environment for all agents
        Reset accumulators
      else
        For each done agent, reset the environment of that agent
        Increment episode count for that agent
        Reset accumulators for that agent
      end if
    end if
    $s \leftarrow s'$
  end for
  After rollout, compute GAE:
  Reward normalization r_i using (28)
  Compute returns δ_i, advantages A_i, and normalize advantages $\hat{A}_i$ for all experiences in buffer according to (20)–(22) respectively
  MAPPO update:
  for epoch $= 1$ to K **do**
    for minibatch in buffer **do**
      Compute policy ratio ρ_i, PPO surrogate loss J_i, and value loss $L_{V_{\phi_i}}$ as defined in (24), (23), (27) respectively
      Compute policy entropy using (25), then multiply by entropy coefficient c_{ent}
      Sum entropy bonus with policy loss as defined in (26), and then add value loss to obtain the total loss
      Backpropagate, update network parameters (shared across agents)
    end for
  end for
  Clear the rollout buffer
  Reset states for all agents
end for

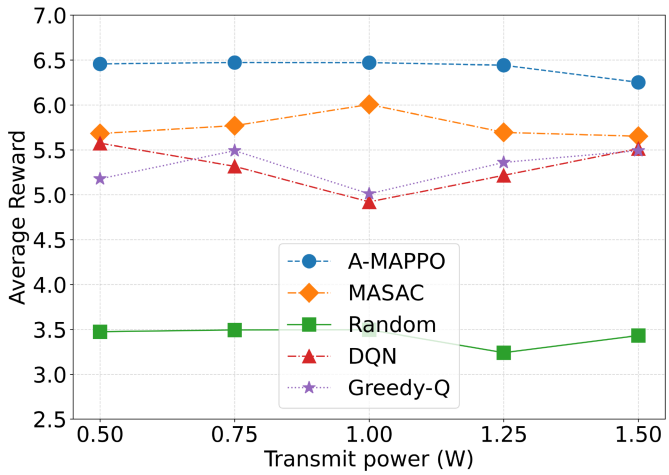

Fig. 7. Average total rewards varying transmit power.

It can be observed that the variation in the average total reward is mainly influenced by the extrinsic reward, while the intrinsic reward remains nearly constant as the task payoff increases, since task payoff appears only in the extrinsic reward term in (11). For A-MAPPO, the intrinsic reward fluctuates within the range of approximately $2.8 - 2.9$ across all payoff values. MASAC exhibits a similar stable behavior, with intrinsic reward remaining around $2.9 - 3.1$ with minor fluctuations. Similar stability is observed for DQN and Greedy-Q, with intrinsic reward remaining around $2.2 - 2.4$, while the Random policy stays constant at about 0.9 as shown in Fig. 6(c).

Since task payoff is only associated with extrinsic reward, higher task payoff directly affects extrinsic reward and consequently the total reward. As shown in Fig. 6(e), A-MAPPO again achieves the strongest performance, with extrinsic reward increasing from approximately 2.4 at a payoff of 5 to nearly 13.8 at a payoff of 25. MASAC also exhibit similar trend, increasing from around 2.2 to approximately 12.3, remaining competitive but consistently below A-MAPPO. DQN and Greedy-Q also follow similar upward trends but at lower levels, reaching about 12.1 and 11.0, respectively, at the highest payoff, which are about 12.3% and 20.3% lower than A-MAPPO.

5) Impact of Transmit Power: Fig. 7 illustrates the impact of transmit power on the average total reward as the transmit power varies from 0.5 W to 1.5 W when $N = 10$. According to (11), increasing the transmit power is expected to raise the communication cost, which would generally lead to lower reward. However, a higher transmit power simultaneously improves the achievable data rate, compensating for the increased communication cost. As shown in Fig. 7, these opposing effects largely

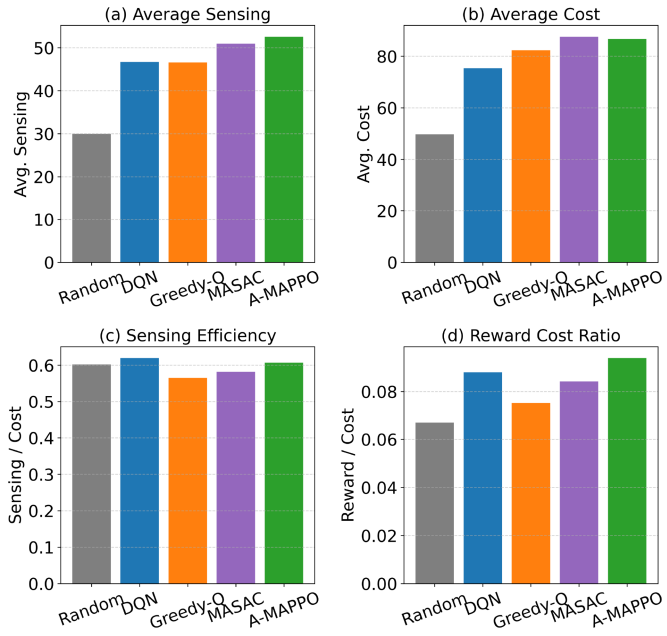

Fig. 8. Average sensing, reward, sensing/cost ratio, and reward/cost ratio comparison.

balance each other, resulting in an almost constant average reward across all models as the transmit power increases. For instance, A-MAPPO maintains a nearly constant average reward of around 6.5, whereas DQN and Greedy-Q policy exhibit minor fluctuations within the range of 5.0 to 5.5. MASAC also maintains a relatively stable performance across different transmit power levels, remaining close to 5.7-6.0 with only slight fluctuations. The random baseline consistently yields a significantly lower reward, remaining around 3.5.

6) Normalized Cost: During asynchronous episodes with variable length, agents collect different amounts of sensing data and incur different battery costs. In this setting $N = 15$, the A-MAPPO collects the highest average sensed data per episode at approximately 54 unit, outperforming Random (≈ 30), DQN (≈ 46), Greedy-Q policy (≈ 46), and MASAC (≈ 51) as shown in Fig. 8(a), but it also incurs a sensing cost of approximately 86 units per episode, ranking as the second highest among all methods, just below MASAC, which reaches 88 units as shown in Fig. 8(b).

When normalizing by sensing cost, the sensing-efficiency metric (sensing per unit battery cost) in Fig. 8(c) shows that all learned policies exhibit comparable performance within a narrow range, with DQN achieving the highest sensing efficiency at 0.62, followed closely by A-MAPPO at 0.61. In contrast, the reward-cost ratio (reward obtained per unit battery cost) in Fig. 8(d) demonstrates that A-MAPPO attains the highest value ($\approx .10$), confirming its ability to extract more reward from each unit of battery consumption compared to DQN ($\approx .09$), MASAC ($\approx .084$), Greedy-Q policy ($\approx .078$), and Random ($\approx .068$). This suggests that while DQN optimizes raw sensing efficiency marginally better, A-MAPPO achieves higher reward-generating outcomes which directly aligns with our primary objective of incentivizing vehicle participation in VCS campaigns.

VI. Conclusion

In this paper, we investigated a satellite-assisted VCS architecture that enables large-scale sensing in scenarios with limited terrestrial infrastructure. We developed a comprehensive incentive mechanism that jointly accounts for immediate sensing rewards, sensing cost, and communication cost under vehicle–satellite channel conditions. In addition, intrinsic rewards arising from vehicular social effects were incorporated. Within this framework, vehicles independently optimize their sensing levels to maximize long-term individual reward. To address the inherent asynchronicity of practical VCS systems arising from vehicles completing their sensing tasks at different times due to heterogeneous task completion or battery constraint, we proposed an asynchronous MAPPO algorithm under a CTDE paradigm. The proposed approach effectively handles heterogeneous episode lengths and decentralized decision-making while leveraging global information during centralized training. Extensive numerical results demonstrate that the proposed A-MAPPO consistently outperforms baseline schemes in terms of achieving highest cumulative reward. This results validate the effectiveness of the proposed framework, highlighting its potential for scalable and efficient satellite-assisted VCS deployments. The current study considers a single-satellite-assisted VCS system with simplified channel modeling. Future work will extend the framework to a multi-satellite system incorporating Doppler effects, and adopt a more realistic satellite-to-vehicle channel model accounting vehicle mobility.

References

[1] Y. Fu, X. Qin, X. Zhang, and Y. Jia, "Hybrid recruitment scheme based on deep learning in vehicular crowdsensing," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 10735–10748, Oct. 2023.

[2] Y. Yu, X. Xue, J. Ma, S. Zhang, Y. Guan, and R. Lu, "Achieving efficient and privacy-preserving worker selection with arbitrary spatial ranges for vehicular crowdsensing," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 13765–13776, Sep. 2024.

[3] Y. Zhao and C. H. Liu, "Social-aware incentive mechanism for vehicular crowdsensing by deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2314–2325, Apr. 2021.

[4] J. Xue et al., "Sparse mobile crowdsensing for cost-effective traffic state estimation with spatio–temporal transformer graph neural network," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 16227–16242, May 2024.

[5] C. Hu et al., "Digital twin-assisted real-time traffic data prediction method for 5G-enabled internet of vehicles," *IEEE Trans. Industr. Inform.*, vol. 18, no. 4, pp. 2811–2819, Apr. 2022.

[6] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 772–782, Jun. 2017.

[7] K. Kim, S. Cho, and W. Chung, "HD map update for autonomous driving with crowdsourced data," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1895–1901, Apr. 2021.

[8] C. Kim, S. Cho, M. Sunwoo, P. Resende, B. Bradaï, and K. Jo, "Updating point cloud layer of high definition (HD) map based on crowd-sourcing of multiple vehicles installed lidar," *IEEE Access*, vol. 9, pp. 8028–8046, 2021.

[9] P. Dai, X. Wang, Y. Xiang, X. Wu, J. Wang, and K. Liu, "Multi-agent reinforcement learning for freshness-aware data sensing model in vehicular crowdsensing systems," *IEEE Trans. Service Comput.*, vol. 18, no. 4, pp. 2212–2225, Jul./Aug. 2025.

[10] J. Huo, L. Wang, Z. Lu, and X. Wen, "Vehicular crowdsensing inference and prediction with multi training graph transformer networks," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 217–227, Jan. 2024.

[11] X. Chen et al., "PAS: Prediction-based actuation system for city-scale ridesharing vehicular mobile crowdsensing," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3719–3734, May 2020.

[12] 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 Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2853–2862, Mar. 2022.

[13] X. Cai et al., "An incentive mechanism for vehicular crowdsensing with security protection and data quality assurance," *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 9984–9998, Aug. 2023.

[14] Y. Cheng, J. Ma, Z. Liu, Y. Wu, K. Wei, and C. Dong, "A lightweight privacy preservation scheme with efficient reputation management for mobile crowdsensing in vehicular networks," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 3, pp. 1771–1788, May/Jun. 2022.

[15] P. Singh, B. Hazarika, K. Singh, W.-J. Huang, and C.-P. Li, "Augmented multiagent DRL for multi-incentive task prioritization in vehicular crowdsensing," *IEEE Internet Things J.*, vol. 11, no. 22, pp. 36140–36156, Nov. 2024.

[16] M. Franke, R. Stroop, F. Klingler, and C. Sommer, "Low earth orbit satellite supported multi-hop dissemination of messages in V2X networks," in *Proc. IEEE Veh. Technol. Conf.*, Florence, Italy, Jun. 2023, pp. 1–5.

[17] P. Lonergan, "Advances in direct to device emergency communications networks," European Emergency Number Association (EENA), Brussels, Belgium, Tech. Rep., Mar. 2025, [Online]. Available: https://eena.org/knowledge-hub/documents/advances-in-direct-to-device-emergency-communications-networks/

[18] X. Chang, M. S. Obaidat, C. Xu, J. Ma, X. Xue, and Y. Yu, "Dynamic federated learning with differential privacy for complex vehicular crowdsensing systems under mobility consideration," *IEEE Syst. J.*, vol. 19, no. 3, pp. 860–871, Sep. 2025.

[19] X. Chang, M. S. Obaidat, J. Ma, and X. Xue, "Grid-assisted federated learning in vehicular crowdsensing: A mobility-aware and probabilistic vehicle selection approach," *IEEE Trans. Veh. Technol.*, vol. 74, no. 7, pp. 11264–11280, Jul. 2025.

[20] G. Yao, J. Huo, L. Wang, Z. Lu, L. Liu, and X. Wen, "Predictive recruitment in vehicular crowdsensing based on spatial sensing strength analysis method," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 3159–3176, Mar. 2024.

[21] Y. Lai, Y. Xu, D. Mai, Y. Fan, and F. Yang, "Optimized large-scale road sensing through crowdsourced vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3878–3889, Apr. 2022.

[22] Y. Zhao, C. H. Liu, T. Yi, G. Li, and D. Wu, "Energy-efficient ground-air-space vehicular crowdsensing by hierarchical multi-agent deep reinforcement learning with diffusion models," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 12, pp. 3566–3580, Dec. 2024.

[23] A. Chakeri, X. Wang, Q. Goss, M. I. Akbas, and L. G. Jaimes, "A platform-based incentive mechanism for autonomous vehicle crowdsensing," *IEEE Open J. Intell. Transp. Syst.*, vol. 2, pp. 13–23, 2021.

[24] L. G. Jaimes, H. Chintakunta, and P. Abedin, "SenseNow: A time-dependent incentive approach for vehicular crowdsensing," *IEEE Open J. Intell. Transp. Syst.*, vol. 5, pp. 307–321, 2024.

[25] L. Liu, X. Wen, L. Wang, Z. Lu, W. Jing, and Y. Chen, "Incentive-aware recruitment of intelligent vehicles for edge-assisted mobile crowdsensing," *IEEE Internet Things J.*, vol. 69, no. 10, pp. 12085–12097, Oct. 2020.

[26] H. Yu, Y. Yang, H. Zhang, R. Liu, and Y. Ren, "Reputation-based reverse combination auction incentive method to encourage vehicles to participate in the VCS system," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2469–2481, Jul.–Sep. 2021.

[27] Q. Cao, Z. Li, H. Li, S. Zhou, and Y. Zhang, "Participant recruitment of vehicular crowdsensing along freeways for traffic accident detection," *IEEE Trans. Mobile Comput.*, vol. 24, no. 10, pp. 9650–9663, Oct. 2025.

[28] X. Guo, X. Wang, W. Pan, W. Wu, and K. Liu, "A Lyapunov optimization approach for long-term task assignment in vehicular crowdsensing," *IEEE Trans. Consum. Electron.*, vol. 71, no. 1, pp. 2107–2117, Feb. 2025.

[29] Y. Yu, X. Xue, J. Ma, E. Z. Zhang, Y. Guan, and R. Lu, "Efficient privacy-preserving task allocation with secret sharing for vehicular crowdsensing," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9473–9486, Mar. 2024.

[30] G. Sun, S. Sun, H. Yu, and M. Guizani, "Toward incentivizing fog-based privacy-preserving mobile crowdsensing in the internet of vehicles," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4128–4142, May 2020.

[31] C. Xiang et al., "LSTAloc: A driver-oriented incentive mechanism for mobility-on-demand vehicular crowdsensing market," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 3106–3122, Apr. 2024.

[32] M. He, H. Wu, X. Shen, and W. Zhuang, "Cooperative resource scheduling for environment sensing in satellite–terrestrial vehicular networks," *IEEE Internet Things J.*, vol. 12, no. 6, pp. 6734–6748, Mar. 2025.

[33] L. Wang, J. Li, M. Dai, and H. Zhang, "Low-earth-orbit satellite assisted edge computing for vehicular networks: A task priority-based delay minimization approach," *IEEE Internet Things J.*, vol. 12, no. 17, pp. 35482–35496, Sep. 2025.

[34] W. Zhan et al., "Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5449–5465, Jun. 2020.

[35] M. Li, J. Gao, L. Zhao, and X. Shen, "Deep reinforcement learning for collaborative edge computing in vehicular networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 4, pp. 1122–1135, Dec. 2020.

[36] J. Yan, X. Zhao, and Z. Li, "Deep-reinforcement-learning-based computation offloading in UAV-assisted vehicular edge computing networks," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19882–19897, Jun. 2024.

[37] L. Liu, J. Feng, X. Mu, Q. Pei, D. Lan, and M. Xiao, "Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15513–15526, Dec. 2023.

[38] P. Li, Z. Xiao, X. Wang, K. Huang, Y. Huang, and H. Gao, "EPtask: Deep reinforcement learning based energy-efficient and priority-aware task scheduling for dynamic vehicular edge computing," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 1830–1846, Jan. 2024.

[39] Z. Wei, B. Li, R. Zhang, X. Cheng, and L. Yang, "Many-to-many task offloading in vehicular fog computing: A multi-agent deep reinforcement learning approach," *IEEE Trans. Mobile Comput.*, vol. 23, no. 3, pp. 2107–2122, Mar. 2024.

[40] Y. Ju et al., "NOMA-assisted secure offloading for vehicular edge computing networks with asynchronous deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 3, pp. 2627–2640, Mar. 2024.

[41] A. Nassar and Y. Yilmaz, "Deep reinforcement learning for adaptive network slicing in 5G for intelligent vehicular systems and smart cities," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 222–235, Jan. 2022.

[42] H. Zhang, S. Chong, X. Zhang, and N. Lin, "A deep reinforcement learning based D2D relay selection and power level allocation in mmWave vehicular networks," *IEEE Wirel. Commun. Lett.*, vol. 9, no. 3, pp. 416–419, Mar. 2020.

[43] H. Zhu, Q. Wu, X.-J. Wu, Q. Fan, P. Fan, and J. Wang, "Decentralized power allocation for MIMO-NOMA vehicular edge computing based on deep reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12770–12782, Jul. 2022.

[44] Y. Li, L. Feng, M. Mei, A. Ali, and Z. Shah, "Joint optimization for volatile federated learning in vehicular edge computing: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 74, no. 9, pp. 14632–14644, Sep. 2025.

[45] S. Moon and Y. Lim, "Client selection for federated learning in vehicular edge computing: A deep reinforcement learning approach," *IEEE Access*, vol. 12, pp. 131337–131348, 2024.

[46] Y. Ju et al., "Energy-efficient cooperative secure communications in mmWave vehicular networks using deep recurrent reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 10, pp. 14460–14475, Oct. 2024.

[47] Q. Wu, Y. Zhao, Q. Fan, P. Fan, J. Wang, and C. Zhang, "Mobility-aware cooperative caching in vehicular edge computing based on asynchronous federated and deep reinforcement learning," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 66–81, Jan. 2023.

[48] N. Aung et al., "VeSoNet: Traffic-aware content caching for vehicular social networks using deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8638–8649, Aug. 2023.

[49] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 247–257, Jan. 2020.

[50] Y. Pan, Z. Su, Y. Wang, J. Zhou, and M. Mahmoud, "Privacy-preserving byzantine-robust federated learning via deep reinforcement learning in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 74, no. 6, pp. 9461–9475, Jun. 2025.

[51] T. Cai et al., "Cooperative data sensing and computation offloading in UAV-assisted crowdsensing with multi-agent deep reinforcement learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3197–3211, Sep./Oct. 2021.

[52] T. Du, X. Gui, and T. Sheng, "Multiagent deep reinforcement learning-based hierarchical scheduling in heterogeneous UAV-enabled vehicular networks," *IEEE Internet Things J.*, vol. 12, no. 24, pp. 54938–54954, Dec. 2025.

[53] C. Zhu, Y.-H. Chiang, Y. Xiao, and Y. Ji, "FlexSensing: A QoI and latency-aware task allocation scheme for vehicle-based visual crowdsourcing via deep Q-network," *IEEE Internet Things J.*, vol. 8, no. 9, pp. 7625–7637, May 2021.

[54] X. Zhu, Y. Luo, A. Liu, N. N. Xiong, M. Dong, and S. Zhang, "A deep reinforcement learning-based resource management game in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2422–2433, Mar. 2022.

[55] Y. Hou, Z. Wei, R. Zhang, X. Cheng, and L. Yang, "Hierarchical task offloading for vehicular fog computing based on multi-agent deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 3074–3085, Apr. 2024.

[56] J. Hu, L. Chen, S. Shen, and T. Wang, "Explainable multiagent deep reinforcement learning for joint task offloading and resource allocation in distance and channel-aware NOMA vehicular edge networks," *IEEE Internet Things J.*, vol. 12, no. 24, pp. 53505–53519, Oct. 2025.

[57] J. Tian, Q. Liu, H. Zhang, and D. Wu, "Multiagent deep-reinforcement-learning-based resource allocation for heterogeneous QoS guarantees for vehicular networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1683–1695, Feb. 2022.

[58] K. Zhang, J. Cao, and Y. Zhang, "Adaptive digital twin and multiagent deep reinforcement learning for vehicular edge computing and networks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1405–1413, Feb. 2022.

[59] H. Wu, B. Wang, H. Ma, X. Zhang, and L. Xing, "Multiagent federated deep-reinforcement-learning-based collaborative caching strategy for vehicular edge networks," *IEEE Internet Things J.*, vol. 11, no. 14, pp. 25198–25212, Jul. 2024.

[60] S. B. Prathiba, G. Raja, S. Anbalagan, A. KS, S. Gurumoorthy, and K. Dev, "A hybrid deep sensor anomaly detection for autonomous vehicles in 6G-V2X environment," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 3, pp. 1246–1255, May/Jun. 2023.

[61] Y. Liang, H. Wu, and H. Wang, "Asynchronous multi-agent reinforcement learning for collaborative partial charging in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4806–4818, May 2024.

[62] B. Yoo, Y. Shin, H. Kim, E. Chung, and J. Yang, "Adaptive episode length adjustment for multi-agent reinforcement learning," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, Detroit, MI, USA, Jun. 2025, pp. 2253–2261.

[63] Y. Xiao, W. Tan, and C. Amato, "Asynchronous actor-critic for multi-agent reinforcement learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, New Orleans, LA, Nov. 2022, pp. 4385–4400.

[64] J. Yin, W. Rao, Y. Xiao, and K. Tang, "Cooperative path planning with asynchronous multiagent reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 24, no. 6, pp. 5016–5030, Jun. 2025.

[65] A. Srinivasan, J. Zhang, and O. Tirkkonen, "Asynchronous multi-agent reinforcement learning for scheduling in subnetworks," in *Proc. IEEE Veh. Technol. Conf.*, Oslo, Norway, Jun. 2025, pp. 1–6.

[66] A. A. R. Alsaeedy and E. K. P. Chong, "A survey of mobility management in non-terrestrial 5G networks: Power constraints and signaling cost," *IEEE Access*, vol. 12, pp. 107529–107551, 2024.

[67] S. C. Prabhashana et al., "Machine learning-based resource allocation in 6G integrated space and terrestrial networks-aided intelligent autonomous transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 10, pp. 17750–17762, Oct. 2025.

[68] K. Wei, Q. Tang, J. Guo, M. Zeng, Z. Feng, and Q. Cui, "Resource scheduling and offloading strategy based on LEO satellite edge computing," in *Proc. IEEE Veh. Technol. Conf. (VTC2021-Fall)*, Norman, OK, USA, Sep. 2021, pp. 1–6.

[69] K. K. Nguyen, S. R. Khosravirad, D. B. D. Costa, L. D. Nguyen, and T. Q. Duong, "Reconfigurable intelligent surface-assisted Multi-UAV networks: Efficient resource allocation with deep reinforcement learning," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 3, pp. 358–368, Apr. 2022.

[70] X. Lin et al., "Intelligent adaptive MIMO transmission for nonstationary communication environment: A deep reinforcement learning approach," *IEEE Trans. Commun.*, vol. 13, no. 8, pp. 5965–5979, Aug. 2025.

[71] Y. Song, X. Li, H. Ji, and H. Zhang, "Energy-aware task offloading and resource allocation in the intelligent LEO satellite network," in *Proc. IEEE Annu. Int. Symp. Pers.,Indoor Mobile Radio Commun.*, Kyoto, Japan, Sep. 2022, pp. 481–486.

[72] P. Erd6s and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hungar. Acad. Sci*, vol. 5, pp. 17–61, 1960.

[73] T. H. T. Truong, P. Seiler, and L. E. Linderman, "Analysis of networked structural control with packet loss," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 1, pp. 344–351, Jan. 2022.

[74] Y. Zhan, Y. Xia, and J. Zhang, "Quality-aware incentive mechanism based on payoff maximization for mobile crowdsensing," *Elsevier Ad Hoc Networks*, vol. 72, pp. 44–55, Jan. 2018.

[75] M. H. Cheung, F. Hou, and J. Huang, "Make a difference: Diversity-driven social mobile crowdsensing," in *Proc. IEEE Conf. Comput. Commun.*, Atlanta, GA, May 2017, pp. 1–9.

[76] J. Nie, J. Luo, Z. Xiong, D. Niyato, and P. Wang, "A Stackelberg game approach toward socially-aware incentive mechanisms for mobile crowdsensing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 724–738, Jan. 2019.

[77] Z. Xiong, D. Niyato, P. Wang, Z. Han, and Y. Zhang, "Dynamic pricing for revenue maximization in mobile social data market with network effects," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1722–1737, Mar. 2020.

[78] T. B. Kirigin and S. B. Babić, "Semi-local integration measure for directed graphs," *Mathematics*, vol. 12, no. 7, Apr. 2018.

[79] X. Ning, M. Zeng, M. Hua, and Z. Fei, "Multiple reconfigurable intelligent surfaces aided vehicular edge computing networks: A MAPPO-based approach," *IEEE Trans. Veh. Technol.*, vol. 73, no. 11, pp. 17496–17509, Nov. 2024.

[80] R. Chattopadhyay and C.-K. Tham, "Joint sensing and processing resource allocation in vehicular ad-hoc networks," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 616–627, Jan. 2023.

[81] L. Xiao, Y. Li, G. Han, H. Dai, and H. V. Poor, "A secure mobile crowdsensing game with deep reinforcement learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 1, pp. 35–47, Jan. 2018.