Ensuring Threshold AoI for UAV-Assisted Mobile Crowdsensing by Multi-Agent Deep Reinforcement Learning With Transformer

Hao Wang, Chi Harold Liu, *Senior Member, IEEE*, Haoming Yang, Guoren Wang, and Kin K. Leung, *Fellow, IEEE*

***Abstract*— Unmanned aerial vehicle (UAV) crowdsensing (UCS) is an emerging data collection paradigm to provide reliable and high quality urban sensing services, with age-of-information (AoI) requirement to measure data freshness in real-time applications. In this paper, we explicitly consider the case to ensure that the attained AoI always stay within a specific threshold. The goal is to maximize the total amount of collected data from diverse Point-of-Interests (PoIs) while minimizing AoI and AoI threshold violation ratio under limited energy supplement. To this end, we propose a decentralized multi-agent deep reinforcement learning framework called "DRL-UCS(AoI$_{th}$)" for multi-UAV trajectory planning, which consists of a novel transformer-enhanced distributed architecture and an adaptive intrinsic reward mechanism for spatial cooperation and exploration. Extensive results and trajectory visualization on two real-world datasets in Beijing and San Francisco show that, DRL-UCS(AoI$_{th}$) consistently outperforms all nine baselines when varying the number of UAVs, AoI threshold and generated data amount in a timeslot.**

***Index Terms*— UAV crowdsensing, AoI, multi-agent deep reinforcement learning, transformer.**

I. Introduction

MOBILE crowdsensing (MCS [1], [2], [3], [4]) has become a promising paradigm to facilitate city sensing applications, such as road monitoring [5], ensuring mobile connectivity [6] and emergency preparedness and response [7]. Different from traditional human-centric MCS, utilizing multiple unmanned aerial vehicles (UAVs) equipped with high-speed data receivers can provide long-term and time-sensitive sensing services. Thus, increasing research efforts emerge on UAV crowdsensing (UCS), where UAVs are deployed to collect sensory data from multiple Point-of-Interest (PoIs) like CCTV cameras and detective sensors. The advantage of UCS is that UAVs can reduce the local network overload by collecting data through reliable communication channels. Furthermore, UAVs can be deployed rapidly and patrolled flexibly in inaccessible zones, which is hardly possible for traditional human-centric crowdsensing paradigm. Take emergency response for example, once happened, the patrolling UAVs are able to immediately supply monitoring data collected from various PoIs with flexible motions, which is critical in extreme situations like rural fire and earthquake.

Data freshness is key to the success of a UCS task, e.g., in emergency response, data recording the occurrence and progress of emergency situations will devalue and no longer be "fresh" if the emergency subsides. To this end, Kaul et al. introduced the concept of episodic "age-of-information" (AoI [8]) as a metric that measures the freshness of collected data. However, in extreme cases, such as disaster response [9], there exists a specific "AoI threshold" that represents the maximum delay tolerance for each data. Once exceeded, data will devalue much faster (e.g., life data of people trapped under earthquake ruins; after 72 hours, they may face severe life risks). In these scenarios, it is important to minimize total AoI as well as maintain them under a prescribed AoI threshold. Thus we proposed a new metric called "threshold AoI" to jointly consider the effect of AoI and the specific predetermined threshold by applications.

To this end, in this paper, we study the problem of navigating multiple UAVs under limited energy supplements in a large-scale workzone (whose typical size can be tens of quare kilometers), collecting data from deployed PoIs to maximize data collection ratio as well as minimizing threshold AoI. Fig. 1 shows an example of the considered AoI threshold aware UCS task, where data are consistently generated by PoIs, and UAVs are navigated to collect them in a collaborative manner. Key challenges are, first, UAVs should carefully plan their trajectories over a long-time horizon according to the time-varying data generation speed and limited energy supplies. Second, UAVs should learn a collaborative movement pattern, like division of work, to maximize their individual sensing capability spatiotemporally. Third, UAVs need to carefully balance the trade-off between data freshness and AoI threshold violation. This is because minimizing total episodic AoI (i.e., keeping data as fresh as possible) might overlook some outlying PoIs and the certain PoI's AoI may surpass its threshold; on the other hand, minimizing total AoI threshold

Manuscript received 17 December 2022; revised 22 May 2023; accepted 20 June 2023; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. P. Jue. Date of publication 12 July 2023; date of current version 16 February 2024. This work was supported by the National Natural Science Foundation of China under Grant U21A20519 and Grant 61772072. *(Corresponding author: Chi Harold Liu.)*

Hao Wang, Chi Harold Liu, Haoming Yang, and Guoren Wang are with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: chiliu@bit.edu.cn).

Kin K. Leung is with the Electrical and Electronic Engineering (EEE) Department, and the Computing Department, Imperial College, SW7 2BT London, U.K. (e-mail: kin.leung@imperial.ac.uk).

Digital Object Identifier 10.1109/TNET.2023.3289172

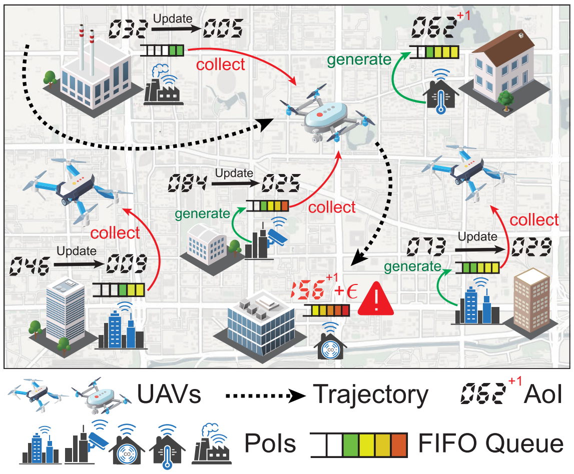


Fig. 1. Overview of the considered AoI threshold aware UCS.

violation ratio neglects data freshness which leads to sub-optimal overall AoI and data collection ratio. Two sides are equally important.

The considered problem can be formulated as a constrained trajectory planning problem, which is fundamentally different from classical Vehicles Routing Problems (VRP [10]) that no specific position needs to be visited in our scenario, and UAVs can collect data from nearby PoIs without moving quite closed-by. We opt to achieve multiple objectives simultaneously, including minimizing threshold AoI under limited energy reserve while maximizing data collection ratio. Given the rapid expansion of the solution space with increasing number of UAVs and task duration, traditional optimization algorithms face significant challenges in searching the optimal solution. Existing heuristic methods are usually assumed to know a precise optimization model based on prior expert knowledge [11] and global information for model iteration, which is infeasible in practice. Considering that the interaction between UAVs and PoIs conforms the Markov property, we model it as a sequential decision problem and leverage Deep Reinforcement Learning (DRL) methods to solve it.

DRL has achieved unprecedented success in many control applications like Go [12] and OpenAI Five [13]. Some existing works used DRL to train a powerful deep neural network to learn the complex moving patterns for UAVs [14], [15], [16]. The most intuitive way to use it is to build a centralized controller with global observation and then control all UAVs simultaneously. However, it is not applicable in real-world because of the high dependency between UAVs and base stations in real-time communication. To make matters worse, the computational cost and the action space will be dramatically expanded with the increasing of UAV numbers and making it hard to find an optimal solution for single-agent DRL algorithms. However, using multi-agent DRL (MADRL [17]) algorithms can effectively decompose the action space, leading to a much smaller action space for each agent. Consequently, it is much easier for each agent to find their own near-optimal policy. Thus we formulate the problem under a MADRL decentralized setting where UAVs are running individually without specific communication assumptions. However, it is quite challenging to directly apply any existing decentralized MADRL methods because of the complex optimization problem, such as how to explore a trajectory properly. Our contribution is three-fold:

- We propose a new metric called "Threshold AoI" to explicitly measure to what extent the AoI threshold is maintained. We propose an objective decomposition technique through where the optimization problem is reformulated as an equivalent sequential decision making problem.
- We propose a decentralized MADRL framework called "DRL-UCS(AoI_{th})" that consists of a transformer (GTrXL)-enhanced distributed MADRL architecture for temporal modeling, and a random network distillation (RND) controlled intrinsic reward mechanism for both spatial cooperation and exploration.
- We perform extensive experiments on two real-world datasets in Beijing and San Francisco, and find the most appropriate hyperparameters. Results confirm that DRL-UCS(AoI_{th}) outperforms all nine baselines in terms of threshold AoI, episodic AoI and AoI threshold violation ratio.

The rest of the paper is organized as follows. In Section II, we review related works. Section III presents the system model. In Section IV, we propose DRL-UCS(AoI_{th}). We show several experimental results in Section V, followed by a conclusion in Section VI.

II. Related Work

A. UAV Crowdsensing (UCS)

Due to the high mobility, flexible deployment and adaptability of UAVs, UAV crowdsensing has been considered as an enabling technology for many applications [18], [19], [20]. On the one hand, UAVs can be deployed as aerial base stations for wireless communication coverage or to improve the connectivity of ground users [21]. On the other hand, UAVs can act as airborne wireless relays, extending the sensing range for inaccessible area [22]. Liu et al. [23] investigated routing multiple UAVs as aerial base stations, to collect data from PoIs to provide reliable sensing services. Luo et al. [24] proposed a fine-grained trajectory planning problem for multi-UAVs and design the detailed hovering and traveling plans on their paths to collect data from sensor networks. Wang et al. [25] studied the problem of utilizing multiple UAVs as communication relays to offload computation tasks and receive sensing result. Zhong et al. [26] proposed a UAVs-aided self-organized network and achieve a high capacity via the joint optimization of relay deployment, channel allocation, and relay assignment. However, most of these mentioned studies did not take the data freshness into consideration.

There also some works that considered data freshness optimization in UAV crowdsensing. Yuan et al. [27] considered an AoI optimization problem with i.i.d. and Markovian data arrivals by giving a closed-form solution. Dai et al. [28] extended it to more realistic settings with trajectory optimization and AoI minimization with constrained energy for multi-antenna UAVs in a fixed sensor network. However, these works only considered linearly scaled data freshness with time. Kaul et al. [8] introduced a general age penalty function to characterize the level of dissatisfaction on data freshness under generate-at-will settings. Xu et al. [9] studied a minimum UAV deployment problem instead of path planning to ensure the freshness of the collected data. Nevertheless, rare work has studied a metric that jointly considers continuously generated data with AoI threshold aware data freshness.

B. MADRL

In MADRL, each agent has to make its individual decisions based on local observations. Directly applying policy improvement methods by considering other agents as part of the environment offers poor convergence property. To address this, a learning paradigm named centralized training with decentralized execution (CTDE) was developed. In CTDE, each agent is equipped with a joint value function which, during training, has access to the global state and opponents' actions. With the help of the centralized value function that accounts for the non-stationarity caused by others, each agent adapts its policy parameters accordingly. Consequently, many multi-agent policy gradient algorithms have been developed, such as QMIX [29], MADDPG [30] and MAPPO [31]. However, in many real-world scenarios, a central controller that learned a centralized value function simply does not exist or may be costly to install. Moreover, the central controller must communicate with each agent to exchange information, which incessantly increases the communication overhead at the single controller. This may degrade the scalability of the multi-agent system as well as its robustness.

To handle policy training in decentralized settings, Schröder proposed IPPO [32], a multi-agent variant of proximal policy optimization (PPO [33]), using policy clipping help mitigate some forms of environment non-stationarity. Zhang et al. [34] considered the problem that decentralized agents are connected via a time-varying and possibly sparse communication network. Jiang et al. [35] proposed a momentum-based decentralized policy gradient tracking, to approximate the local policy gradient surrogate with importance sampling. However, both of them are trained and converged slowly due to the lack of scalability and low efficiency in collecting experiences. The state-of-the-art approach for distributed DRL is IMPALA [36] which played hundreds of Atari games and reached the top league of StarCraft II without any game restrictions. We use IMPALA as the start point of our design, but it is still quite challenging to adopt it into a decentralized MADRL framework for an AoI threshold aware UCS task.

C. Trajectory Planning Related Solutions

VRP is a typical combinatorial optimization problem to find optimal routing trajectories, to deliver packages from a depot to customers while minimizing the total travel cost. It has numerous variants that address different routing subproblems. For example, the Capacitated VRP (CVRP [37]) involves vehicles with limited capacity; the VRP with Time Windows (VRPTW [38]) requires that each customer be served within a specific time window, and the Multiple Depot VRP (MDVRP [39]) involves supplying customers from multiple depots. When using deep neural networks (DNNs) to solve VRP problems, there are typically two approaches: construction methods that directly output solutions [40], [41], and improvement methods that assist with local search [42], [43]. However, the problem considered in this paper is a multi-objective trajectory planning problem and it is ineffective to solve it by using the point-to-point access pattern of VRP-based path planning.

Another line of possible solutions is to transform the above optimization problem into a sequential decision-making problem and seek to solve it using DRL. Along this direction, Liu et al. [44] proposed a DQN-based trajectory design algorithm for a single UAV-mounted MEC network to ensure the quality of service of each user. Liu et al. [45] proposed a DDQN-based framework for reconfigurable intelligent surfaces (RIS)-enhanced sensing service in single UAV-enabled wireless networks. Zhang et al. [46] leveraged a MADDPG-based approach to maximize the secure capacity by jointly considering trajectory design and power optimization of UAVs. Zhao et al. [47] developed a centralized MATD3-based framework for multi-UAVs trajectory design in dynamic multi-access edge computing environments. However, the centralized MADRL framework brings additional communication overhead in practical applications. Also, as the problem complexity and solution space increase when deploying more UAVs, a well-designed intrinsic reward mechanism is critical to guide DRL agents for effective and reasonable trajectory planning, which is a key contribution of this paper as RND controlled intrinsic reward in Section IV-B.

III. System Model

We assume that a group of UAVs $\mathcal{U} \triangleq \{u|u = 1, 2 \ldots, U\}$ are moving around in the workzone, to collect data from several PoIs $\mathcal{P} \triangleq \{p|p = 1, 2, \ldots, P\}$ that generate data continuously (e.g., surveillance cameras continuously monitor the environment in the dense traffic area). In emergency situations where wired communication might be disrupted, PoIs can transmit the data packet to a UAV wirelessly. In addition, there exist obstacles (e.g., skyscrapers and no-fly zones) that UAVs cannot go through. Without loss of generality, we model the air-to-ground propagation channel by jointly considering the Line-of-Sight (LoS) and NLoS (non-Line-of-Sight) effects, along with their occurrence probabilities respectively. Following [48], the probability of LoS connections between a UAV u and a PoI p is given by:

$$\mathrm{Pr}_{LoS}^{u,p} = \frac{1}{1 + \alpha_1 \exp\left(-\alpha_2(\vartheta^{u,p} - \alpha_1)\right)},$$

where α_1 and α_2 are constants which depend on the environment (i.e., rural or urban). $\vartheta^{u,p}$ denotes the elevation angle. Therefore, the large scale pathloss effect between a UAV u and a PoI p is calculated by:

$$\varphi_t^{u,p} = \mathrm{Pr}_{LoS}^{u,p} \times PL_{LoS} + \mathrm{Pr}_{NLoS}^{u,p} \times PL_{NLoS}.$$

Here, PL_{Los} and PL_{NLoS} denote the average path loss for LoS and NLoS links, respectively.

$$PL_{LoS} = 20\log(d^{u,p}) + 20\log(4\pi\alpha_F) + \alpha_{LoS},$$
$$PL_{NLoS} = 20\log(d^{u,p}) + 20\log(4\pi\alpha_F) + \alpha_{NLoS},$$

TABLE I
IMPORTANT NOTATIONS USED IN THIS PAPER

Notation	Explanation
$u, U, \mathcal{U}$	Index, total number, set of UAVs
$p, P, \mathcal{P}$	Index, total number, set of PoIs
$t, T, \tau, \tau^u_{t,m}, \tau^u_{t,c}$	Index, total number of timeslots, duration of a timeslot, duration for UV movement and data collection in each timeslot
$\alpha_1, \alpha_2, \alpha_F$	Constants depend on environment settings
$\mathrm{Pr}^{u,p}_{LoS}, \mathrm{Pr}^{u,p}_{NLoS}$	Probability of LoS and NLoS connections
PL_{LoS}, PL_{NLoS}	Average path loss for LoS and NLoS links
$\varphi^{u,p}_{t,snr}, \varphi^{u,p}_t$	Received SNR, transmission power and the large scale fading
$B, \varphi_{tx}, \varphi_n$	Transmission bandwidth, transmission power and average noise power
δ^p_t, Δ	Current data amount, generated data amount in a timeslot
$\mathrm{rate}^{u,p}_t, \mathcal{I}^{u,p}_t$	data transmission rate and data collected indices
v_{move}, v_{tip}, v_0	UAV movement speed, tip speed, rotor speed
$E_{\max}, \alpha_{coll}, \alpha_{move}$	UAV initial energy, energy consumption coefficient for data collection and UAV movement
c_1, c_2, c_3	Constants depend on power, rotors, air density
$X, m^p(\cdot)$	Task duration, generation time of the oldest data at PoI p
$w^p, \mathcal{V}^p, \epsilon$	weight of a PoI p, set of time when AoI threshold is violated at PoI p, and violation penalty
$x^p_{i,g}, x^p_{i,c}$	Data i's generation and collection time at PoI p
$\kappa, \omega, \chi, \zeta, \xi$	Threshold AoI, episodic AoI, AoI threshold violation ratio, data collection ratio, energy consumption ratio
$\boldsymbol{o}_t, \boldsymbol{a}_t, r_t$	Observation, action and reward over all UAVs in timeslot t

where $d^{u,p}$ is the Euclidean distance, a_F is a constant depending on channel frequency and the speed of light, α_{LoS} and α_{NLoS} are average additional path loss from environment. Note that we assume that data upload transmissions from different PoIs will not interfere with each other by using OFDMA in 802.11ax [49]; other complicated communications models can also be used but it is not the focus of this paper. Given the fixed transmission power φ_{tx} and average noise power φ_n, we can calculate the received signal-to-noise-ratio (SNR) by $\varphi^{u,p}_{t,snr} = \varphi_{tx} - \varphi^{u,p}_t - \varphi_n$. Following [50], we assume that the received SNR must exceed a SNR threshold φ_0, otherwise the transmission will fail. By using Shannon Capacity, the maximum transmission rate between a UAV u and a PoI p becomes $\mathrm{rate}^{u,p}_t = B\log(1+\varphi^{u,p}_{t,snr})$, where B is the bandwidth.

Our considered UCS tasks work as follows. A task is divided into T equivalent timeslots of length τ, i.e., $\{0, 1, \ldots, t, \ldots, T-1\}$ where t is the index of the current timeslot. At the beginning, U UAVs are deployed at the same origin and each PoI p initializes with data volume $\delta^p_0 = 0$. Each PoI p either generates a fixed-size data packet Δ (e.g., a picture with fixed resolution) or not in each timeslot, thus the current data amount is denoted by $\delta^p_t = \delta^p_{t-1} + \Delta$. Each PoI also maintains a First-In First-Out (FIFO) queue and data leave the queue only when UAVs approach and collect it.

Each UAV u spends $\tau^u_{t,m}$ amount of time to move in a certain direction $\vartheta^u_t \in (0, 2\pi)$ at a fixed speed. During the remaining time $\tau^u_{t,c} = \tau - \tau^u_{t,m}$, UAVs will collect data from at most $P_{\max}$ nearest PoIs following a round robin policy. Without loss of generality, we assume that PoI transmits a complete packet each time, and if the current packet is not transmitted successfully, the PoI will re-transmit it in the next collection time. After, the remaining data at PoI p becomes $\delta^p_t - \Delta\sum_u |\mathcal{I}^{u,p}_t|$, where $\mathcal{I}^{u,p}_t$ is the set that represents data indices collected by UAV u, satisfying $\sum_p(|\mathcal{I}^{u,p}_t|\Delta/\mathrm{rate}^{u,p}_t) \leq \tau^u_{t,c}$.

It is worth noting that each UAV has limited energy reserve $E_{\max}$. The task fails immediately if any UAV runs out of energy. The energy consumption of a UAV u in timeslot t is given by $e^u_t = \alpha_{move} * \tau^u_{t,m} + \alpha_{coll} * \tau^u_{t,c}$. Following [51], the energy consumption coefficient α is calculated by:

$$\alpha = c_1\left(1 + \frac{3v^2_{\mathrm{uav}}}{v^2_{tip}}\right) + c_2\left(\sqrt{1 + \frac{v^4_{\mathrm{uav}}}{4v^4_0}} - \frac{v^2_{\mathrm{uav}}}{2v^2_0}\right)^{1/2} + \frac{1}{2}c_3 v^3_{\mathrm{uav}}, \tag{1}$$

where v_{uav} represents the UAV speed, $v_{\mathrm{uav}} = v_{move}$ and $v_{\mathrm{uav}} = 0$ when calculating α_{move} and α_{coll}, respectively. The constants c_1, c_2, c_3 depend on power, rotors and air density of each UAV. v_{tip} and v_0 denote the tip speed of the rotor blade and the average velocity induced by the rotor, respectively.

A. Performance Metrics

To begin with, we will introduce all metrics that we consider in this paper. As we have mentioned, we are particularly interested in the freshness of data, and we want all the data get collected in some given time periods, if not continually but still at the end all of them. However, as UAVs are patrolling, it is impractical to set hard constraints and achieve no violations, hence we consider a new metric with soft constraint penalty applied to AoI whenever it goes over a given threshold.

Definition 1 (Episodic AoI): Let $m^p(x) \in \mathcal{R}$ be a random process in $x \in [0, X]$ representing the generation time of the oldest data at a PoI p, where $X = T \cdot \tau$. Hence term $(x - m^p(x))$ represents the waiting time before it is collected which is considered as episodic AoI [8]:

$$\omega = \frac{1}{PX}\sum_{p=1}^{P} w^p \int_0^X x - m^p(x)\,dx, \tag{2}$$

where weights w^p of each PoI p are introduced to indicate the importance of different various locations (e.g, some are associated with heavy traffic while others are with generally light traffic).

Definition 2 (AoI Threshold Violation Ratio): Let $\mathcal{V}^p$ denotes the timeslot set of AoI threshold is violated at PoI p, i.e., $\mathcal{V}^p = \{x | x \in [0, X], x - m^p(x) \geq \mathrm{AoI}_{th}\}$, then χ is computed as:

$$\chi = \frac{1}{PX}\sum_{p=1}^{P}\int_0^X \mathbb{1}_{\mathcal{V}^p}(x)\,dx,$$

where $\mathbb{1}$ is the indicator function.

Definition 3 (Threshold AoI): Suppose mapping function f is a transformation applied to AoI, Γ is the soft constrained penalty function, then Threshold AoI is written as:

$$\kappa = \frac{1}{PX}\sum_{p=1}^{P} w^p \int_0^X f(x - m^p(x))\, dx, \tag{3}$$

where:

$$f(x - m^p(x)) = \left\{ \begin{matrix} \Gamma(x - m^p(x)), \text{ if } x \in \mathcal{V}^p \\ x - m^p(x), \quad \text{otherwise} \end{matrix} \right\}. \tag{4}$$

Meanwhile, Γ needs to satisfy several properties: (a) $\Gamma(x - m^p(x)) \geq x - m^p(x), \forall x \in \mathcal{V}^p$, which guarantees that penalty is positive. (b) Γ needs to be non decreasing on each connected interval, which makes sure that penalty are non decreasing with time when AoI goes over the threshold. (c) Γ needs to be measurable and has origin function so that integration is well defined.

Definition 4 (Data Collection Ratio): let ζ denotes the total collection ratio, then:

$$\zeta = \frac{\sum_{p=1}^{P}\sum_{t=0}^{T-1} \Delta|\mathcal{I}_t^p|}{\sum_{p=1}^{P}(\delta_{T-1}^p + \sum_{t=0}^{T-1} \Delta|\mathcal{I}_t^p|)}. \tag{5}$$

B. Problem Definition

Our goal is to jointly minimize threshold AoI and maximize collection ratio with energy consumption no more than $E_{\max}$. As depicted in Fig. 2(a), we consider the penalty function as a constant penalty in this paper, i.e., $\Gamma(x) = x + \epsilon$, if $x \in \mathcal{V}^p$, and x otherwise. Then, our overall objective becomes:

$$\min_{m} \frac{1}{PX}\sum_{p=1}^{P} w^p \left\{ \int_0^X x - m^p(x)\, dx + \int_{\mathcal{V}^p} \epsilon\, dx - \sum_{t=0}^{T-1} \Delta|\mathcal{I}_t^p| \right\} \tag{6}$$

$$\text{s.t.} \sum_{t=0}^{T-1} e_t^u \leq E_{\max}, \quad \forall u \in \mathcal{U}. \tag{7}$$

Proposition 1: If $\Gamma(x) = x + \epsilon$, then minimizing threshold AoI is equivalent to jointly minimizing episodic AoI and AoI threshold violation ratio.

Proof:

$$\begin{aligned} \kappa &= \frac{1}{PX}\sum_{p=1}^{P} w^p \int_0^X f(x - m^p(x))\, dx \\ &= \frac{1}{PX}\sum_{p=1}^{P} w^p \left\{ \int_{\mathcal{V}^p} \Gamma(x - m^p(x))\, dx + \int_{\bar{\mathcal{V}}^p} x - m^p(x)\, dx \right\} \\ &= \frac{1}{PX}\sum_{p=1}^{P} w^p \left\{ \int_{\mathcal{V}^p} x - m^p(x) + \epsilon\, dx + \int_{\bar{\mathcal{V}}^p} x - m^p(x)\, dx \right\} \\ &= \frac{1}{PX}\sum_{p=1}^{P} w^p \left\{ \int_0^X x - m^p(x)\, dx + \int_{\mathcal{V}^p} \epsilon\, dx \right\} \end{aligned}$$

■

The above proposition indicates that jointly minimizing episodic AoI and threshold violation ratio is a special case of minimizing threshold AoI.

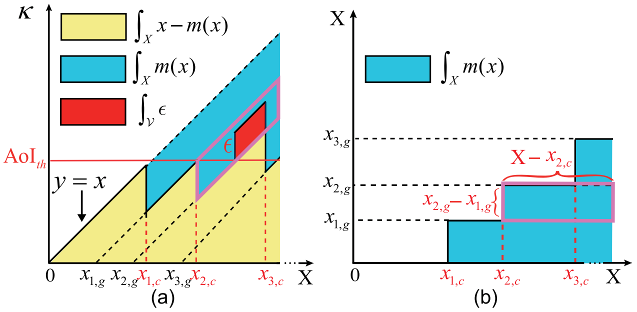


Fig. 2. Proposed threshold AoI κ with ϵ-constant penalty function. For simplicity, we omit the superscript which represents PoI p. The blue area represents the integration of $m(x)$ and the red area denotes the integration $\int_\mathcal{V} \epsilon$. To calculate the average κ, one can either integrate areas below the curve or compute the isosceles right triangle bounded by $y = x$, plus the penalty area, and then subtract with the integration of $m(x)$. Both of the above computations give the same result.

Lemma 1: For a given PoI p, term $\int_0^X m^p(x)$ has equivalent form as:

$$\int_0^X m^p(x)\, dx \;= \sum_{i \in \mathcal{I}_t^p} (x_{i,g}^p - x_{i-1,g}^p)(X - x_{i,c}^p), \tag{8}$$

where $x_{i,g}^p$ and $x_{i,c}^p$ denote the generated and collected time of data i at PoI p.

Proof: We illustrate $\int_0^X m^p(x)$ as the blue area in Fig. 2. One can either compute this integral horizontally or vertically. The decomposition above is computing horizontally. ■

Proposition 2: Then, the objective has an equivalent step-by-step computational decomposition property, as:

$$\begin{aligned} &\frac{1}{X} w^p \left\{ \int_0^X x - m^p(x)\, dx + \int_{\mathcal{V}^p} \epsilon\, dx - \sum_{t=0}^{T-1} \Delta|\mathcal{I}_t^p| \right\} = \frac{1}{X} w^p \\ &\sum_{t=0}^{T-1} \left\{ \frac{X^2}{2} - \sum_{i \in \mathcal{I}_t^p} (x_{i,g}^p - x_{i-1,g}^p)(X - x_{i,c}^p) + \epsilon \cdot |\mathcal{V}_t^p| - \Delta|\mathcal{I}_t^p| \right\}, \end{aligned} \tag{9}$$

where $\mathcal{V}_t^p = \{x | x \in [t * \tau, (t+1) * \tau], x - m^p(x) \geq \text{AoI}_{th}\}$ and satisfing $\bigcup_{t=0}^{T-1} \mathcal{V}_t^p = \mathcal{V}^p$.

Proof: Eqn. (6) can be written as:

$$\frac{w^p}{X} \left\{ \int_0^X x - \int_0^X m^p(x) + \int_0^X \epsilon \cdot \mathbb{1}_{\mathcal{V}^p}(x) - \sum_{t=0}^{T-1} \Delta|\mathcal{I}_t^p| \right\}. \tag{10}$$

The first term is $\frac{X^2}{2}$, the second term can be substituted by Lemma 1 and the third term can be written as $\int_0^X \epsilon \cdot \mathbb{1}_{\mathcal{V}^p}(x) = \sum_{t=1}^{T} \epsilon \cdot |\mathcal{V}_t^p|$. It is equivalent to Eqn. (9) if we pulled the summation on t out. ■

Lemma 2: Assuming that each UAV can service all PoIs without movement and instantaneously collect data from $P_{\max}$ PoIs in a timeslot, then the AoI lower bound $\omega_{\min}$ can be calculated by:

$$\omega_{\min} = \frac{1}{PX}\sum_{p=1}^{P} w^p \frac{X^2}{2} - \sum_{t=0}^{T-1} \sum_{p \in \mathcal{P}_t} S_t^p,$$

where $S_t^p = w^p \sum_{i \in \mathcal{I}_t^p} (X - t\tau)(x_{i,g}^p - x_{i-1,g}^p)$, $\mathcal{P}_t$ is the collection set comprising $UP_{\max}$ PoI indices and satisfying:

$$S_t^p \geq S_t^{\hat{p}}, \forall p \in \mathcal{P}_t, \bar{p} \in \bar{\mathcal{P}}_t, \ \mathcal{P}_t \sqcup \bar{\mathcal{P}}_t = \mathcal{P}.$$

Proof: As shown in Fig. 2, S_t^p represents the obtained blue region when collecting all current data from PoI p at timeslot t, $\mathcal{P}_t$ represents the collection of top $UP_{\max}$ PoIs with largest S_t^p at timeslot t. Meanwhile, minimizing AoI is strictly equivalent to maximizing the area of blue region, which is determined by the outcome of data collection at current timeslot and independent of future data arrivals. Therefore, we can greedily select PoIs that contribute to a larger blue area for collection to obtain the AoI lower bound. ■

The above optimization problem is NP-Hard since even checking the optimally requires a thorough search of the entire trajectory space with exponential amountal of elements with respect to timeslot T and the number of UAV U. Hence, a good heuristic trajectory planning strategy is required. Based on Proposition 2, our considered problem can be naturally modeled as a sequential decision problem and utilizing DRL methods to solve it.

C. Problem Formulation as Dec-POMDP

To solve above sequential decision problem, we first formulate it as a decentralized partially observable Markov decision problem (Dec-POMDP) denoted as a 7-tuple $M = < \mathcal{U}, \mathcal{O}, \mathcal{A}, R, \Omega, \gamma >$, where Ω and γ stand for transition probabilities and discounted factor, respectively.

1) Observation: observation space $\mathcal{O} \triangleq \{\boldsymbol{o}_t\}$. Each UAV maintains its local observation $\boldsymbol{o}_t^u$ with a fixed sensing range, which can be written in a disjoint union of $\boldsymbol{o}_t^u(\text{PoI})$ and $\boldsymbol{o}_t^u(\text{UAV})$. The former contains the location, remaining data amount and data generation time for all PoIs within the sensing range; and the latter contains current position and remaining energy for all UAVs during training.

2) Action: action space $\mathcal{A} \triangleq \{\boldsymbol{a}_t\}$. For each UAV, $\boldsymbol{a}_t^u$ is a 2-tuple: (ϑ_t^u, l_t^u), where ϑ_t^u represents the angle which controls the direction of UAV movement and l_t^u is the traveling distance, which is bounded by a maximum distance $l_{\max}$.

3) Reward function: based on Proposition 2, we have the following reward from the environment for each UAV:

$$r_{t,\text{env}}^u = \sum_{p=1}^{P} \frac{w^p}{X} \Bigg\{ \sum_{i \in \mathcal{I}_t^{u,p}} (x_{i,g}^p - x_{i-1,g}^p)(X - x_{i,c}^p) \tag{11}$$

$$- \ \epsilon \cdot |\mathcal{V}_t^p| + \Delta |\mathcal{I}_t^p| \Bigg\} + h_t^u. \tag{12}$$

Here h_u^t is the penalty that considers energy consumption or hitting obstacles. It is worth noting that maximizing Eqn. (12) in this Dec-POMDP settings is strictly equivalent to minimizing Eqn. (6), which is compatible with DRL-based methods as the start point of our design.

IV. Proposed Solution: DRL-UCS(AoI$_{th}$)

We propose a decentralized MADRL method called "DRL-UCS(AoI$_{th}$)" for AoI threshold aware UCS. It consists of a transformer (Gated Transformer-XL, GTrXL [52])-enhanced distributed architecture for temporal modeling, and a random network distillation (RND [53]) controlled intrinsic reward mechanism for spatial cooperation and exploration.

A. Transformer-Enhanced Distributed MADRL Framework for AoI Threshold Aware Multi-UAV Navigation

In most cases, decentralized MADRL policies are hard to learn nor converge slowly. A natural improvement is introducing distributed learning that offers high scalability and data throughput such as IMPALA [36]. However, IMPALA is only designed for single agent and here we explicitly extend it into a multi-agent setting as shown in Fig. 3.

Besides, learning different UAV behaviors leads to a non-stationary environment which causes high variance in value estimators. This phenomenon can be mitigated by using a transformer [52] to capture information both over long time horizons and scale to deal with large amount of experiences, which is naturally compatible with a distributed framework. In addition, discovering and modeling the temporal features inside a workzone (e.g., different data generation times) and the past UAV trajectories (e.g., which PoIs have been collected) is the key to complete the UCS tasks effectively.

Therefore, we propose a transformer-enhanced distributed MADRL framework called "DRL-UCS(AoI$_{th}$)", which follows decentralized training and decentralized execution (DTDE) paradigm without any information sharing. It consists of multiple asynchronous actors working on different GPUs, whose goal is to collect sufficient experiences and send them to the learner, where gradient descent methods are used to update each decentralized policy. As shown in Fig. 3, we add two GTrXL blocks to extract temporal features from past trajectories for each UAV. First, we use two multi-layer perceptions MLP($\cdot$) to embed observation $\boldsymbol{o}_t^u$ into a latent variable $\boldsymbol{z}_t^u$, then we concatenate $\boldsymbol{z}_t^u$ with previous trajectories of length L to form an input for multi-head attention [54] as MHA($\cdot$). In detail, we project current embedding $\boldsymbol{z}_t^u$ into a query matrix $\mathcal{Q}$, and project history embedding $[\boldsymbol{z}_{t-L}^u, \ldots, \boldsymbol{z}_{t-1}^u, \boldsymbol{z}_t^u]$ into a key matrix $\mathcal{Q}$ and a value matrix $\mathcal{V}$, respectively. Several parallel attention operations $softmax(\mathcal{Q}\mathcal{K}^\intercal)\mathcal{V}$ are calculated, and then we concatenate these results and feed forward in an MLP layer as the output of MHA module. Note that we use gating layer Gate($\cdot,\cdot$) to help UAVs look further and focus only on the specific memory. Inspired by GRU [55], we adapt its powerful gating mechanism into an untied activation function, as:

$$\boldsymbol{z} = \sigma(W_1\boldsymbol{y} + W_2\boldsymbol{x} - b_g), \quad \boldsymbol{h} = \sigma(W_3\boldsymbol{y} + W_4\boldsymbol{x}),$$
$$\text{Gate}(\boldsymbol{x}, \boldsymbol{y}) = (1 - \boldsymbol{z}) \odot \boldsymbol{x} + \boldsymbol{z} \odot tanh(W_5\boldsymbol{y} + W_6(\boldsymbol{h} \odot \boldsymbol{x})),$$

where $W.$ and b_g denote different embedding weights and gating bias which control the identity initialization, respectively. The forward propagation in each block is calculated by:

$$\begin{aligned} \hat{\boldsymbol{z}}_t^u &= \text{Gate}(\boldsymbol{z}_t^u, \text{MHA}([\boldsymbol{z}_{t-L}^u, \ldots, \boldsymbol{z}_{t-1}^u, \boldsymbol{z}_t^u])), \\ \hat{\boldsymbol{z}}_t^u &= \text{Gate}(\hat{\boldsymbol{z}}_t^u, \text{MLP}(\hat{\boldsymbol{z}}_t^u)). \end{aligned} \tag{13}$$

To stabilize model training, two layer normalization [56] are added. The outputs of the second transformer module is fed to policy and value network, respectively.

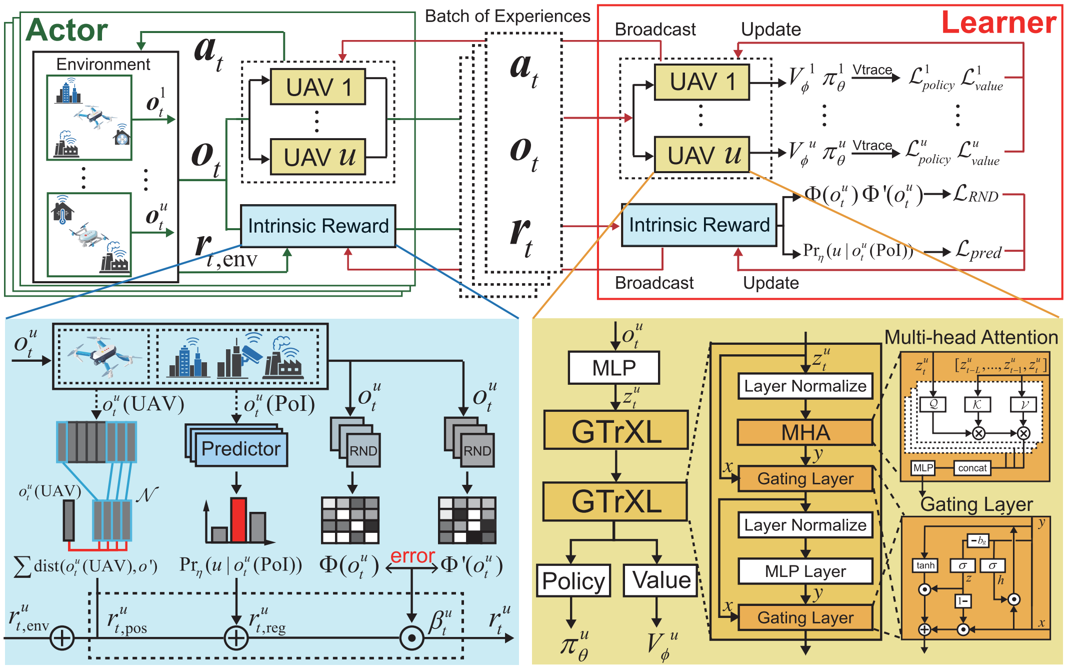


Fig. 3. Proposed Solution: DRL-UCS(AoI$_{th}$).

Each UAV has independent policy as well as value network and they may suffer from different levels of policy lagging and estimation variance. Thus, we calculated the V-trace target for each UAV u to relieves the high variance problem by:

$$V^u(\boldsymbol{o}_t^u) = V_\phi^u(\boldsymbol{o}_t^u) + \sum_{i=t}^{t+n-1} \gamma^{i-t} \left(\prod_{j=t}^{i-1} c_j^u \right) \rho_i^u \text{TD}_i^u, \tag{14}$$

where V_ϕ^u estimates values directly by using the current observation, and TD_i^u is one step temporal difference target $\text{TD}_i^u = r_{i+1}^u + \gamma V_\phi^u(\boldsymbol{o}_{i+1}^u) - V_\phi^u(\boldsymbol{o}_i^u)$, $c_j^u = \min\left(\bar{c}, \frac{\pi_\theta^u(\boldsymbol{a}_j^u|\boldsymbol{o}_j^u)}{\pi_{\theta_{act}}^u(\boldsymbol{a}_j^u|\boldsymbol{o}_j^u)}\right)$, and $\rho_i^u = \min\left(\bar{\rho}, \frac{\pi_\theta^u(\boldsymbol{a}_i^u|\boldsymbol{o}_i^u)}{\pi_{\theta_{act}}^u(\boldsymbol{a}_i^u|\boldsymbol{o}_i^u)}\right)$ are truncated important sampling ratios between the actor's policy θ_{act} and the learner's policy θ. Finally, the optimization objective of the decentralized value network for each UAV is calculated by L-2 loss:

$$\mathcal{L}_{value}^u(\phi) = \mathbb{E}[(V^u(\boldsymbol{o}_t^u) - V_\phi^u(\boldsymbol{o}_t^u))^2]. \tag{15}$$

Therefore, the parameter θ of the decentralized policy network π_θ^u is updated by:

$$\mathcal{L}_{policy}^u(\theta) = \mathbb{E}[\rho_t^u \log \pi_\theta^u(\boldsymbol{a}_t^u|\boldsymbol{o}_t^u)(r_t^u + \gamma V^u(\boldsymbol{o}_t^u) - V_\phi^u(\boldsymbol{o}_t^u))]. \tag{16}$$

Note that policy and value network share part of parameters to accelerate model training.

B. RND Controlled Intrinsic Reward for Simultaneous Spatial Cooperation and Exploration

Sufficient exploration is key to our AoI threshold aware UCS tasks, due to the long decision horizons and exponentially increased search space. Thus we introduce a compound intrinsic reward, computed from spatial diversity to help UAVs explore environment more effectively and cooperatively. We propose RND as a novel exploration adaptor to balance the trade-off between exploration in early stages and experiences exploitation in later stages.

As shown in Fig. 3, we extract two kinds of spatial diversity as intrinsic reward. Intuitively, UAVs need to visit different locations to prevent AoI at certain PoI going over the threshold once in a while. In addition, UAVs should avoid getting too close to each other at the same time which might result in the waste of data collection resources. Since a UAV can only access its own partial observation, we utilize $\boldsymbol{o}_t^u(\text{UAV})$ to encourage positional diversity through spatial K-Nearest neighbors (KNN), and $\boldsymbol{o}_t^u(\text{PoI})$ that composes the status of several nearest PoIs is adopted to predict the likelihood of UAVs that might collect at current PoI. In this way, we make full use of regional diversity to encourage a more straightforward workload division among UAVs.

1) Positional Reward for Spatial Exploration by KNN: We encourage positional diversity by designing KNN-based intrinsic reward. Here we adopt an episodic buffer that stores part of the observations, $\{\boldsymbol{o}_0^u(\text{UAV}), \boldsymbol{o}_1^u(\text{UAV}), \ldots, \boldsymbol{o}_t^u(\text{UAV})\}, \forall u \in \mathcal{U}$. The KNN of $\boldsymbol{o}_t^u(\text{UAV})$ in the episodic buffer are denoted by $\mathcal{N}$, and the location diversity reward is computed as an online fashion by the average distance among them:

$$r_{t,\text{pos}}^u = \frac{1}{|\mathcal{N}|} \sum_{\boldsymbol{o}' \in \mathcal{N}} \text{dist}(\boldsymbol{o}_t^u(\text{UAV}), \boldsymbol{o}'), \tag{17}$$

where $\text{dist}(\cdot)$ is the running average Euclidean distance. Intuitively, positional reward encourages UAVs to visit places where they have not been to recently, since these places might contain PoIs with AoI surpassing threshold.

2) Regional Reward for Spatial Cooperation: Good division of work is a straightforward and effective way of spatial cooperation among UAVs. Each UAV should be responsible for a different region, making the most use of collecting ability while reducing duplication movements. To exploit regional diversity, we use a predictor η to output a conditional

Algorithm 1 Actor

```
Input: policy network $\theta_{act}$, RND network $\Phi_{act}, \Phi'_{act}$,
       predictor $\eta_{act}$ and episodic buffer.
1  while leaner updates do
2      Clear episodic buffer;
3      while episodic buffer is not full do
4          Get UAVs' observation $\boldsymbol{o}_t$ and select actions $\boldsymbol{a}_t$
             from policy $\boldsymbol{\pi}_{\theta_{act}}$;
5          Interact the environment and get reward $\boldsymbol{r}_{t,\text{env}}$;
6          Compute the total rewards $\boldsymbol{r}_t$ by Eqn. (21) and
             stored to experiences buffer;
7      end while
8      Send full episodic buffer to learner;
9      if received broadcast weights $\theta$,$\Phi$,$\Phi'$,$\eta$ then
10         Update networks
             $\theta_{act}, \Phi_{act}, \Phi'_{act}, \eta_{act} \leftarrow \theta, \Phi, \Phi', \eta$;
11     end if
12 end while
```

probability $\text{Pr}_\eta(\mathcal{U}|\mathcal{O})$ for each UAV. Given its observation, each UAV will receive a predicted probability and use it as part of the intrinsic reward at timeslot t, as:

$$r^u_{t,\text{reg}} = \text{Pr}_\eta(u|\boldsymbol{o}^u_t(\text{PoI})). \tag{18}$$

If there is only slight difference among UAVs in terms of their historical observations, then the density $\text{Pr}_\eta(\mathcal{U}|\mathcal{O})$ will have low entropy which leads to a low regional reward. To maximize expected future rewards, UAVs need to learn to achieve spatial division of work with positive feedback provided by regional reward.

To learn a reasonable predictor η, we use an information-theoretic objective to maximize the mutual information between PoI status and UAV identities, as:

$$\begin{aligned}\text{MI}(\mathcal{U};\mathcal{O}) &= \mathcal{H}(\mathcal{U}) - \mathcal{H}(\mathcal{U}|\mathcal{O})\\ &= \mathcal{H}(\mathcal{U}) - \mathbb{E}_{\boldsymbol{o}\sim\text{Pr}_\eta(\boldsymbol{o})}\left[\sum_u -\text{Pr}_\eta(u|\boldsymbol{o})\log\text{Pr}_\eta(u|\boldsymbol{o})\right].\end{aligned}$$

Maximizing the above mutual information can encourage UAVs to discover a new trajectory pattern, as it is easier to infer which UAVs are responsible for the given observation if UAVs' cooperation is more identifiable. Therefore, the parameters η is updated by a self-supervised loss as:

$$\mathcal{L}_{pred}(\eta) = \text{CE}\Big(\text{Pr}_\eta(\cdot|\boldsymbol{o}^u_t(\text{PoI}), \text{OH}(u) * \text{Pr}_\eta(\cdot|\boldsymbol{o}^u_t(\text{PoI}))\Big), \tag{19}$$

where CE and OH denotes cross entropy loss function and one hot function, respectively.

3) Global Adaptor by RND: We also add an exploration adaptor β^u_t to control the degree of spatial exploration. At the beginning of the training stage, we hope our intrinsic reward can be large enough to help UAVs discover interesting positions; yet, as training process proceeds, underlying DNNs become familiar with the environment and hence we want to weaken the impact that intrinsic rewards can make. Nevertheless, it is desirable for UAVs to get higher reward from new observations than those frequently visited regardless of the training progress. Inspired by RND, we propose an embedding network Φ which is trained on previous experiences along with a fixed and randomly initialized network Φ'. We define the exploration adaptor $\beta^u_t = \frac{1}{\beta_{\text{avg}}}\big(\Phi(\boldsymbol{o}^u_t) - \Phi'(\boldsymbol{o}^u_t)\big)^2$, where β_{avg} is the running average. By multiplying β^u_t with intrinsic reward, the consequence of stepping into this new observation will be magnified and quickly reflected on policy networks. Finally, the predictor Φ is trained by gradient descent to minimize the expected L-2 loss after the reward is computed:

$$\mathcal{L}_{RND}(\Phi) = \mathbb{E}\left[\big(\Phi(\boldsymbol{o}^u_t) - \Phi'(\boldsymbol{o}^u_t)\big)^2\right]. \tag{20}$$

We also add an intrinsic coefficient λ to balance magnitude between intrinsic and extrinsic rewards. As a summary, the total reward for UAV u at timeslot t is:

$$r^u_t = r^u_{t,\text{env}} + \lambda \cdot \beta^u_t \cdot (r^u_{t,\text{pos}} + r^u_{t,\text{reg}}). \tag{21}$$

Algorithm 2 Learner

```
Input: policy network $\theta$, value network $\phi$, RND
       network $\Phi, \Phi'$ and predictor $\eta$.
1  Initialize network weights $(\theta, \phi, \Phi, \Phi', \eta)$;
2  while Learner updates do
3      Get experiences from different actor;
4      Compute $\mathcal{L}^u_{value}(\phi), \mathcal{L}^u_{policy}(\theta)$ for each UAV by
         Eqn. (15) and Eqn. (16);
5      Minimizing weighted sum of all losses $\mathcal{L}_{total}$ by
         gradient descent methods;
6      Update $\eta$ and $\Phi$ by Eqn. (19) and Eqn. (20);
7      if updated times mod broadcast interval $= 0$ then
8          Send network weights $\theta, \Phi, \Phi', \eta$ to all actors;
9      end if
10 end while
```

C. Algorithm Description

DRL-UCS(AoI$_{th}$) consists of multiple actors and one learner. Actors are responsible for collecting experiences, and the learner uses these experiences to update DNN weights. The pseudo-code is given in Algorithm 1 and 2.

For each actor, an experience is collected separately and synchronization is not required. Actors simply keep collecting new experiences until the episodic buffer is full; then all actors send these fresh experiences to learner for another update. To be specific, each actor is initialized with a policy network θ_{act}, RND network Φ_{act}, Φ'_{act}, predictor η_{act} and an empty episodic buffer. For each timeslot t, an actor computes the intrinsic reward by Eqn. (21) and stores the experiences into the buffer after all UAVs interact with the environment (Line 4-6). When the experience buffer is full, the actor sends it to the learner (Line 8) and updates its network after receiving broadcast DNN weights (Line 9-11).

For the learner, it updates DNN weights through collected experiences and broadcasts new parameters to all actors. At the beginning, DNN weights (Line 1) is set by Xavier uniform initializer. After gathering a batch of experiences from actors, it computes the $\mathcal{L}^u_{value}(\phi), \mathcal{L}^u_{policy}(\theta)$ according to Eqn. (15)

TABLE II

SIMULATION SETTINGS

Notation	Value	Notation	Value	Notation	Value
T	240	α_1	4.88	φ_n	-104dBm
τ	20s	α_2	0.43	P in BJ	244
$E_{\max}$	719.2KJ	B	20MHz	P in SF	251
v_{move}	20m/s	φ_{tx}	20dBm	Area of BJ	7.8×4.1km^2
v_{tip}	120m/s	φ_0	16dB	Area of SF	6.1×4.4km^2

and Eqn. (16) for each UAV. Next, the learner uses stochastic gradient descent methods to update policy and value estimator by minimizing the weighted sum of above losses (Line 5). Predictor η and RND network Φ will be updated by Eqn. (19) and Eqn. (20), accordingly (Line 6). Note that the learner will send DNN weights to all actors after a period of time (Line 7-9).

It is worth noting that although we utilize decentralized IMPALA as the start point of our design, two proposed contributions can be naturally adopted to other MADRL algorithms (e.g., IPPO [32] and MADDPG [30]. As a powerful temporal feature extractor, GTrXL can provide temporal modeling capability, while RND controlled intrinsic reward represents the additional guided signal for each UAV in the current task, which is compatible with other DTDE algorithms. When extending to CTDE algorithms (e.g., MAPPO [31] and QMIX [29]), one can sum up the intrinsic rewards of all UAVs to represent the extra guided signal for the entire team.

V. EXPERIMENTAL RESULTS

To better reflect the emergency situation however the dataset is not available, we utilize two real-world datasets containing taxi trajectories in Beijing [57] and San Francisco [58] to identify critical areas where people/traffic is highly concentrated to require special sensing service, e.g. during emergency response. Since both are metropolitan cities covering a wide area of landscape thus not suitable for UAVs to fly back and forth within the entire city given their limited power supply, we opt to select two relatively smaller subareas with heavy traffic, and we consider PoIs as surveillance cameras which monitor the top 20% of most frequently visited places. These places are often surrounded by transportation hubs or entertainment centers, where accidents are more likely to occur necessitating faster data sensing services. For example, Beijing Workers Stadium (Fig. 13(a) upper left) and Beijing Railway Station (Fig. 13(a) lower left) in Beijing, Union Square (Fig. 13(d) lower right) in San Francisco, etc. The surveillance area in Beijing is 7.82 kilometers in length and 4.11 kilometers in width, covering an area of about 32.10 squared kilometers with 244 PoIs. And the surveillance area in San Francisco is 6.08 kilometers in length and 4.42 kilometers in width, covering an area of about 6.875 squared kilometers with 251 PoIs. Most of the PoIs in Beijing are distributed on the circular main roads, and only a few are scattered around the corner occasionally. In San Francisco, PoIs are more evenly distributed. Beijing brings more challenges to historical feature extraction, since UAVs need to remember when to visit remote PoI at the cost of a higher AoI. San Francisco needs a proper exploration mechanism to find out the valuable direction when the feedback is similar. We use Google Map to mark the city data, including positions and shapes of tall buildings so that UAVs cannot crash into. Data are generated at each PoI if any taxi passes through. We assign weights to each PoI calculated by normalized visited counts through a consecutive period.

TABLE III

HYPERPARAMETER TUNNING

		Beijing					San Francisco				
λ	L	κ	ω	χ	ζ	ξ	κ	ω	χ	ζ	ξ
0.1	10	35.190	30.439	0.064	0.913	0.782	42.269	36.103	0.089	0.878	0.796
	20	33.186	28.263	0.068	0.937	0.779	37.960	32.621	0.082	0.905	0.821
	30	32.316	28.289	0.056	0.922	0.769	40.210	34.129	0.094	0.896	0.824
	40	39.244	33.238	0.073	0.913	0.764	37.790	31.543	0.094	0.881	0.802
0.3	10	27.236	25.953	0.018	0.964	0.777	37.108	31.660	0.076	0.918	0.809
	20	**26.843**	25.949	0.012	0.969	0.781	31.002	28.058	0.048	0.923	0.812
	30	32.778	29.876	0.038	0.946	0.797	34.502	32.658	0.076	0.906	0.807
	40	35.530	32.528	0.033	0.932	0.789	33.663	28.337	0.081	0.911	0.810
0.5	10	36.170	32.121	0.057	0.910	0.791	31.102	27.458	0.053	0.930	0.808
	20	30.261	28.367	0.025	0.956	0.785	**28.715**	26.133	0.042	0.946	0.801
	30	37.140	32.720	0.056	0.912	0.788	30.685	28.048	0.044	0.936	0.813
	40	39.077	34.959	0.055	0.892	0.789	32.698	29.393	0.053	0.921	0.803
0.7	10	38.235	34.617	0.046	0.916	0.796	44.457	37.771	0.091	0.889	0.814
	20	36.325	33.608	0.030	0.925	0.782	43.911	36.827	0.089	0.891	0.794
	30	47.700	39.595	0.104	0.896	0.799	48.770	40.183	0.119	0.834	0.796
	40	43.840	39.263	0.052	0.878	0.793	48.753	40.886	0.107	0.817	0.815

By investigating parameters in [48] for air-to-ground communications model and referring to the technical report of industrial UAVs like DJI Matrice300 RTK [59]. Simulation settings are summarized in Table II.

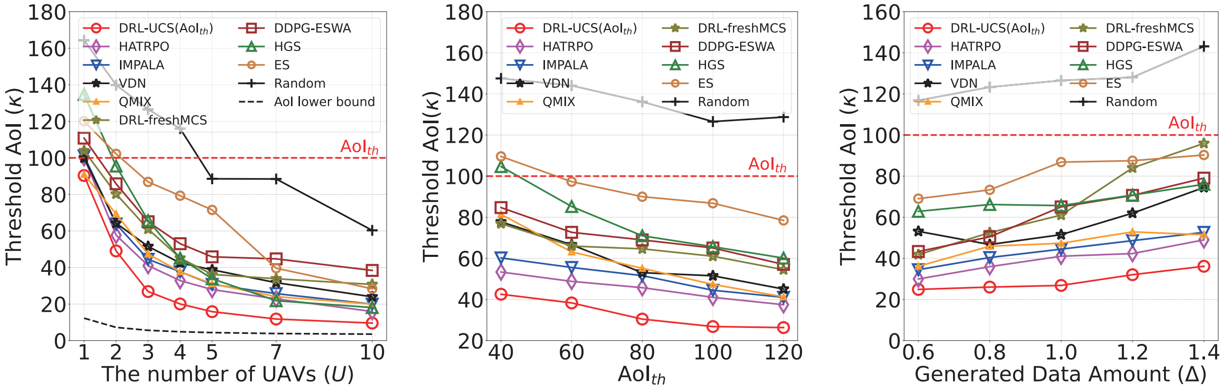
We use the Threshold AoI κ for direct performance comparisons, together with four other metrics: episodic AoI ω, AoI threshold violation ratio χ, data collection ratio ζ and energy consumption ratio $\xi = \frac{1}{U \cdot E_{\max}} \sum_{u=1}^{U} \sum_{t=0}^{T-1} e_t^u$. We consider $U = 3$, AoI$_{th} = 100$ timeslots and $\Delta = 1$Mb. We use Pytorch 1.11.0 to implement our solution, and all codes are run on Ubuntu 18.04.4 LTS with 8 NVIDIA RTX A6000 graphic cards. For testing, we run 50 times on each model and take the average.

A. Hyperparameter Tuning

We first show the results for hyperparameter tuning in DRL-UCS(AoI$_{th}$), as intrinsic coefficient λ (in intrinsic reward) and rollout length L (in GTrXL). We tune $L \in \{10, 20, 30, 40\}$ to study the impact of the observation length that UAVs look back and $\lambda \in \{0.1, 0.3, 0.5, 0.7\}$ to study the effect of percentage of using intrinsic reward. We use 3 heads in each MHA and $b_g = 2$ for gated identity initialization. For predictor η and Φ, we use 3 MLP layers with 256 hidden states and update them by the Adam optimizer with a learning rate 1×10^{-5}. The discount factor γ is 0.99 and batch size is 256. The actors and critic networks are updated by the RMSprop optimizer with a learning rate of 7×10^{-4}. For other hyperparameters, we follow common settings in IMPALA [36] and GTrXL [52]. As shown in Table III, we come up with the following results.

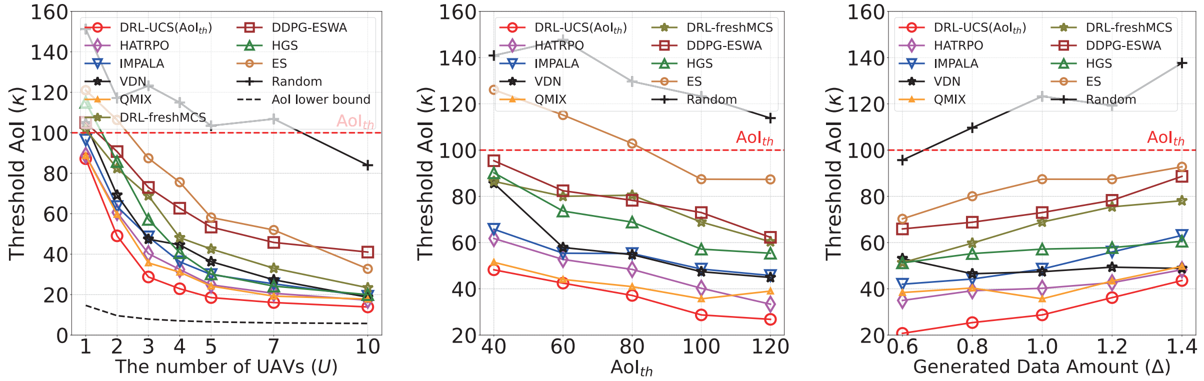
We observe that $L = 20$ yields a peak value in terms of κ with the increase of rollout length L. Looking ahead with a larger L helps UAVs extract a more complete temporal feature and plan their trajectories reasonably. When L is too large, the past embedding may contain redundant information and too many unnecessary details, which is hard to learn and result in a decrease of overall performance significantly.

We found that κ arrived at different peaks when $\lambda = 0.3$ and 0.5, respectively. The reason is ascribed to the difference of



(a) κ (b) κ (c) κ

Fig. 4. Threshold AoI w.r.t. U, AoI_{th} and Δ (Beijing).



(a) κ (b) κ (c) κ

Fig. 5. Threshold AoI w.r.t. U, AoI_{th} and Δ (San Francisco).

TABLE IV
Ablation Study

Dataset	Methods	κ	ω	χ	ζ	ξ
Beijing	DRL-UCS(AoI$_{th}$)	**26.843**	**25.949**	**0.012**	**0.969**	**0.781**
	·· w/o GTrXL	36.432	31.957	0.059	0.895	0.788
	·· w/o intrinsic reward	30.940	27.310	0.050	0.928	0.796
	·· w/o GTrXL & intrinsic reward	44.495	37.791	0.119	0.839	0.804
San Francisco	DRL-UCS(AoI$_{th}$)	**28.715**	**26.133**	**0.042**	**0.946**	**0.801**
	·· w/o GTrXL	36.894	31.832	0.071	0.905	0.820
	·· w/o intrinsic reward	40.951	34.013	0.089	0.881	0.805
	·· w/o GTrXL & intrinsic reward	48.552	40.432	0.102	0.827	0.814

PoI distributions between Beijing and San Francisco. Due to the lack of exploration when λ is small, UAVs will collect data in an inefficient way with an unreasonable division, and may ignore remote PoIs which go over the threshold for a long time. On the other hand, giving too much weights (λ) on the intrinsic reward also do harm to κ, since UAVs will focus more on maximizing intrinsic reward and neglect the original task reward from environment, which might cause a sub-optimal policy.

From Table III, we observe that the set of rollout length $L = 20$, intrinsic reward coefficient $\lambda = 0.3$ and 0.5 perform best in terms of κ in Beijing and San Francisco, respectively; and they will be used for performance comparison hereafter.

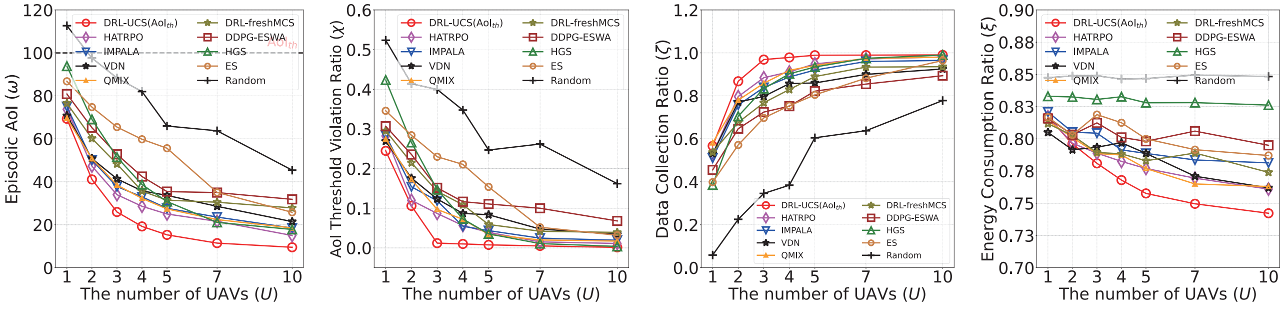
B. Ablation Study

We gradually remove two key components of DRL-UCS(AoI$_{th}$), namely GTrXL and intrinsic reward. The results are shown in Table IV. Compared to DRL-UCS(AoI$_{th}$) without GTrXL, κ achieves 26% and 22% improvements in Beijing and San Francisco datasets, indicating that GTrXL is indeed an indispensable part. GTrXL can accurately remember previous information relevant to the current observation under Dec-POMDP settings. While the intrinsic reward aids in exploring the map to service all PoIs, the lack of modeling and extraction of temporal features in the UAVs prevents them from planning a reasonable path to maintain AoI below the specified threshold. As a result, the removal of GTrXL significantly increased χ from 0.012 to 0.059 in Beijing. When intrinsic reward is removed, κ increases in various degrees, from 26.84 to 30.94 timeslots in Beijing and from 28.71 to 40.95 timeslots in San Francisco. This reveals that our model leverages the exploration skills of intrinsic reward to gain a better result. We can also discover that Beijing takes more advantage of GTrXL because PoIs are separated into two sparsely connected regions; hence more temporal information is required to assist learning procedure. In contrast, San Francisco benefits more from intrinsic reward, since PoIs are more dispersed than those in Beijing. When both GTrXL and intrinsic reward are removed, κ drops significantly, which confirms the benefits of combining intrinsic reward and GTrXL together.

C. Comparing With Nine Baselines

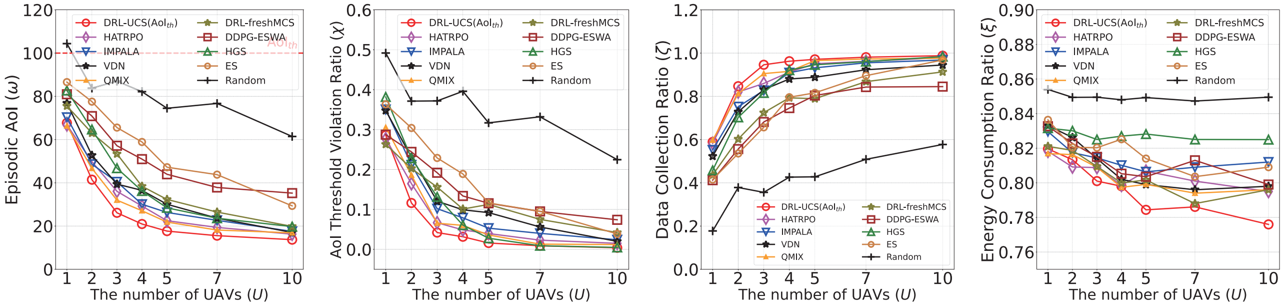
We compare DRL-UCS(AoI$_{th}$) with nine baselines:

- HATRPO [60]: It introduces a multi-agent centralized policy iteration and achieves a guaranteed improvement



(a) ω (b) χ (c) ζ (d) ξ

Fig. 6. Impact of the number of UAVs (Beijing).



(a) ω (b) χ (c) ζ (d) ξ

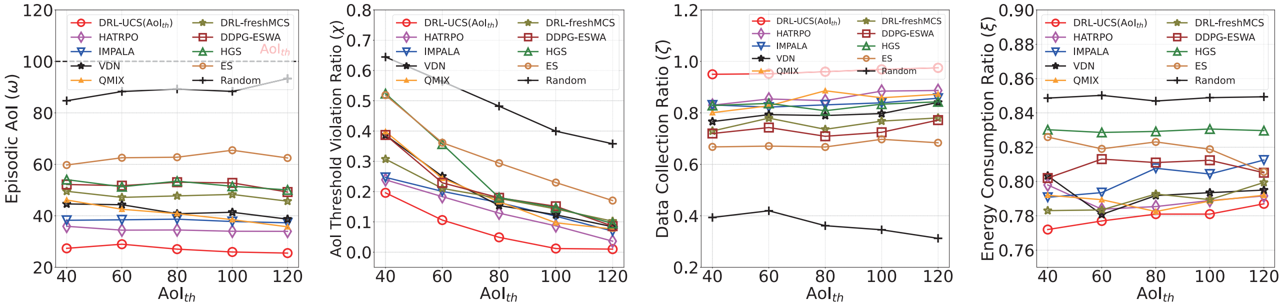
Fig. 7. Impact of the number of UAVs (San Francisco).

without any restrictive assumptions, which is considered as an enhanced version of MAPPO and state-of-the-art centralized MADRL approach.

- IMPALA [36]: It consists of multiple actors and one learner with V-trace target. We adopt it under the DTDE MADRL setting and consider it the state-of-the-art decentralized MADRL approach. Note that IMPALA adds a classical LSTM as a temporal feature extraction module.
- VDN [61]: It is a classical MADRL approach and utilizes a value decomposition network, which learns an optimal linear value decomposition from the global reward to address the credit assignment in cooperative tasks.
- QMIX [29]: It is a classical MADRL approach with an improvement in value decomposition. It utilizes a nonlinear way by learning a mixer network, to allow agents' policies to be updated via individual max operations.
- DRL-freshMCS [62]: It minimizes AoI by navigating a group of UAVs to collect data from deployed sensor nodes. We consider it the state-of-the-art single-agent DRL approach in AoI-aware UCS. To fit the dimension of CNN inputs in DRL-freshMCS, we map UAV observations into a discrete meta image.
- DDPG-EWSA [63]: It is another single-agent DRL-based solution for AoI-aware vehicular networks where UAVs are deployed to process temporally varying traffic streams.
- HGS [11]: It is the state-of-the-art large-scale VRP solver by iteratively identifying and solving smaller subproblems. We model the considered UCS task as a VRP problem, that each UAV follows the shortest path to the location of each PoI and collects data in a sequel.
- Evolution Strategies (ES [64]): It is a traditional optimization algorithm to maximize an objective function by iteratively generating and evaluating a population of candidate solutions. In the implementation, we represent the population as a distribution over parameters and perturb populations with Gaussian noise $\mathcal{N}(0, 0.02)$, then evaluate a population with the objective function defined in Eqn. (6).
- Random: Each UAV u samples action $\boldsymbol{a}_t^u$ from action space $\mathcal{A}$ randomly.

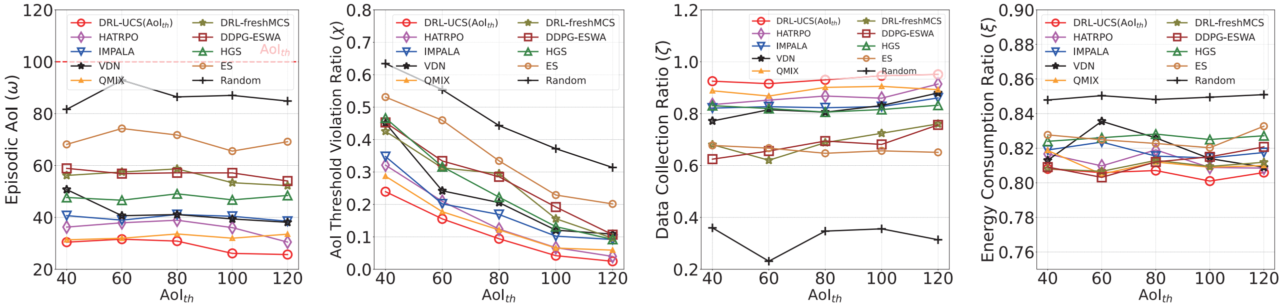
1) Impact of the Number of UAVs: We first show the impact of the number of UAVs by fixing $\text{AoI}_{th} = 100$ timeslots and $\Delta = 1$Mb while varying U from 1 to 10. By introducing GTrXL for temporal modeling and intrinsic reward for a better spatial exploration, DRL-UCS(AoI_{th}) obtains the best κ, ω and χ comparing with all other baselines in both Beijing and San Francisco datasets. For example, in Fig. 4 and Fig. 5, when $U = 3$ DRL-UCS(AoI_{th}) achieves the threshold AoI of 26.84 and 28.71 timeslots in Beijing and San Francisco, compared to 41.05 and 40.24 given by the best baseline HATRPO, with a 35% and 30% improvements.

From Fig. 6 and Fig. 7, we observe that κ and χ decrease while ζ increases with more UAVs. Deploying more UAVs would help share the crowdsensing task by efficient cooperation patterns, which brings lower κ and ξ. But more UAVs also impose the challenge of optimizing policy in exponentially larger solution space. Due to the problem complexity and the large search space involved, ES struggles to find a reasonable trajectory that services various PoIs effectively in long-term decision-making, leading to converging to the local optimum and consequently resulting in poor performance compared to DRL-based algorithms. DDPG-EWSA employed DDPG [65] with Ornstein-Uhlenbeck noise, DRL-freshMCS, VDN and QMIX utilized ϵ-greedy policy to explore which are ineffective in our environment. IMPALA outperforms



(a) ω (b) χ (c) ζ (d) ξ

Fig. 8. Impact of AoI threshold (Beijing).



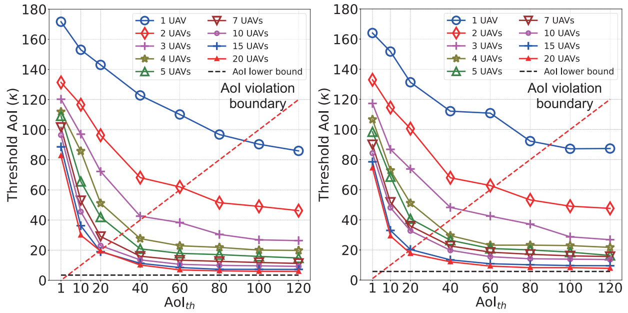
(a) ω (b) χ (c) ζ (d) ξ

Fig. 9. Impact of AoI threshold (San Francisco).

DRL-freshMCS and DDPG-EWSA by a margin, due to its better exploration with stochastic policy, however suffers from serious non-stationary problem due to of the lack of temporal modeling. Besides, we can find that VDN performs worse than QMIX since UAVs' actions mutually influence each other, thus directly summing up Q values does not accurately represent the overall performance and fails to guide policy updates properly. QMIX alleviates this problem by incorporating an additional mixer network during value decomposition, its performance is still inferior to DRL-UCS(AoI$_{th}$) due to the use of ϵ-greedy exploration and the lack of temporal modeling. HATRPO outperforms the rest of baselines but still worse than our proposal DRL-UCS(AoI$_{th}$). This is because HATRPO employs a centralized paradigm which may neglect the harmful behavior brought by a specific UAV since they share the same global reward, which significantly increases the challenge of spatial exploration.

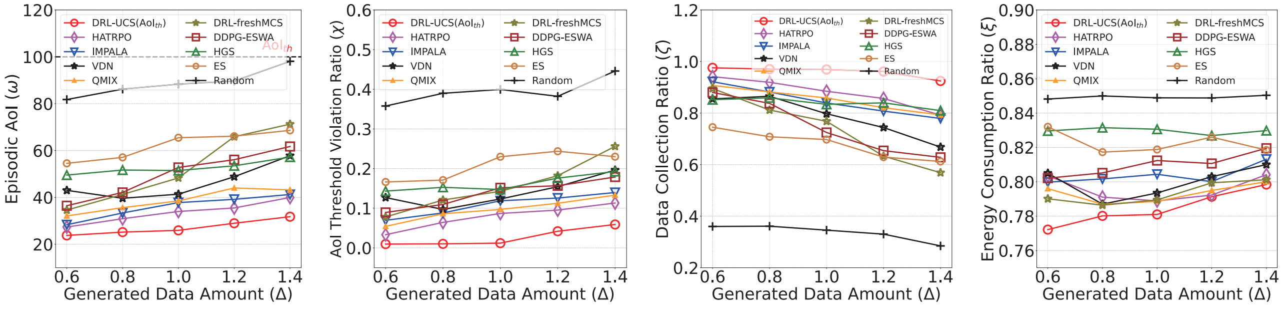
As shown in Fig. 6c and Fig. 7c, HGS collects almost all the data when 7 or more UAVs are deployed; yet, its performance is still worse than DRL-UCS(AoI$_{th}$) in terms of κ, given the fact that it does not consider the relationship between time-varying data generation and spatial location. For example, traffic is heavy in city center during daytime and thus visiting remote areas could result in the increase of κ. A better solution might be visiting remote PoIs when they are about to over AoI threshold. To model the temporal information, we introduced the GTrXL to better use historical experiences and collect data in a more efficient way. Finally, increasing the number of UAVs will obviously reduce the gap between different methods.

2) Impact of AoI Threshold: We fixed $U = 3$ and $\Delta = 1$Mb to investigate how AoI$_{th}$ influences results. As shown in Fig. 8 and Fig. 9, we observe that κ and χ both increase significantly when we lower down AoI$_{th}$ for less stringent data freshness requirement. San Francisco cannot obtain the same level of κ as Beijing under the same AoI$_{th}$, i.e., DRL-UCS(AoI$_{th}$) achieves $\kappa = 42.56, 48.23$ timeslots in Beijing and San Francisco when AoI$_{th} = 40$ timeslots. This is because dispersed PoIs in San Francisco make it hard to explore and find good routing paths for UAVs to maintain a lower AoI for PoIs, while reducing the AoI threshold violation ratio. However, both datasets require strong long-term temporal modeling capability especially when AoI$_{th}$ is very low. Since a UAV needs to assign its limited capability reasonably well and reduce the probability of redundant data collection to deal with more PoIs which might go over the threshold. From Fig. 8b and Fig. 9b, we also observe that the performance gap is gradually reduced between IMPALA and HATRPO due to the memory enhancement by LSTM, but still worse than DRL-UCS(AoI$_{th}$). This is because that the LSTM can not capture long-horizon features through the limited hidden cells, and MHA in GTrXL could extract more precise and



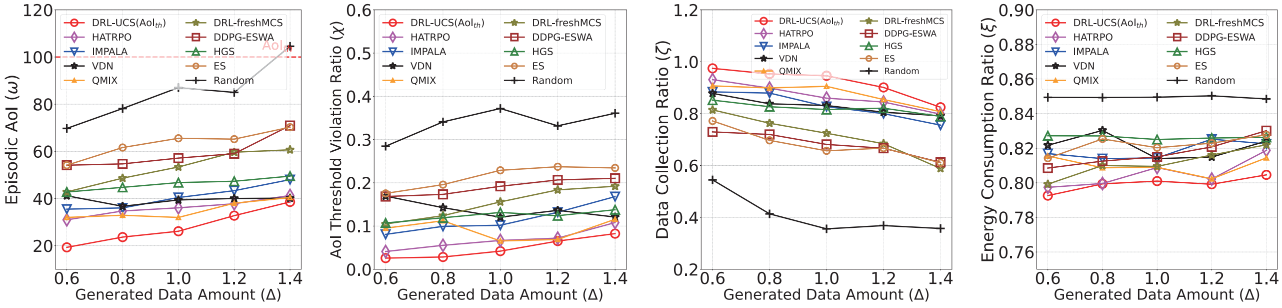
(a) Beijing (b) San Francisco

Fig. 10. Threshold AoI w.r.t. the number of UAVs.



(a) ω (b) χ (c) ζ (d) ξ

Fig. 11. Impact of generated data amount in a timeslot (Beijing).



(a) ω (b) χ (c) ζ (d) ξ

Fig. 12. Impact of generated data amount in a timeslot (San Francisco).

reasonable temporal features for UAVs' path planning by multiple attention operations.

On the other hand, deploying only 3 UAVs is not enough when AoI$_{th}$ is low. Therefore, we show the trend of κ and threshold AoI lower bound $\kappa_{\min}$ when increasing the number of UAVs in Fig. 10. We see that more UAVs could help meet the demand of more stringent data freshness requirements, e.g., when deployed 20 UAVs, DRL-UCS(AoI$_{th}$) achieves $\kappa = 5.905, 7.802$ which are really close to $\omega_{\min} = 3.43, 5.71$ in Beijing and San Francisco, respectively. However, the improvement given by the number of UAVs becomes marginal when $U > 15$.

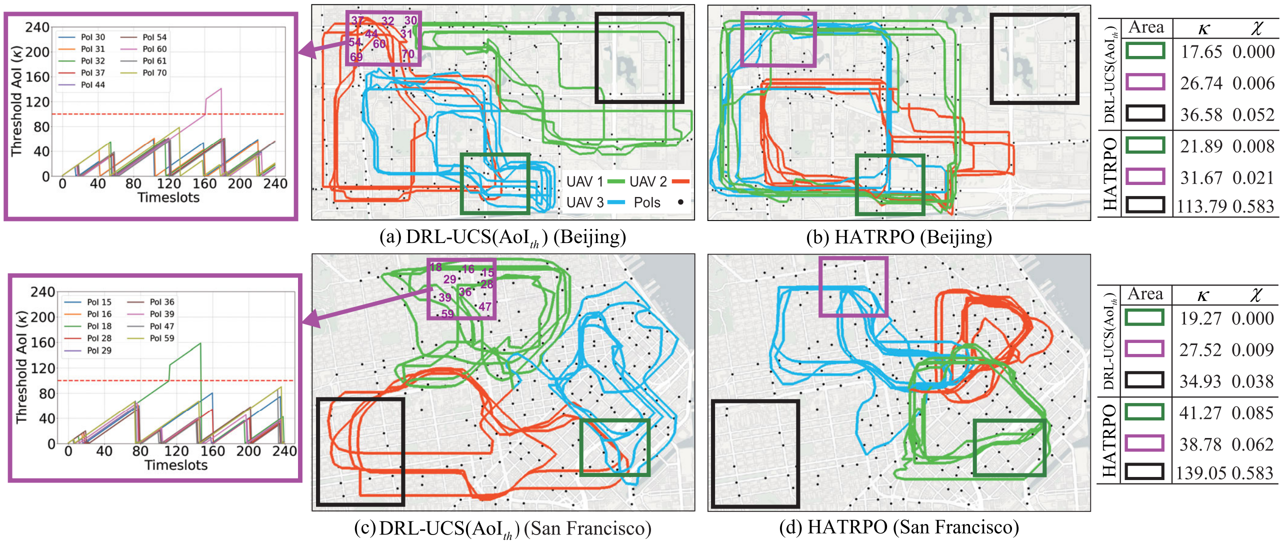
Meanwhile, from Fig. 8(b) and Fig. 9(b), we notice that ω is not monotonically decreasing with the increase of AoI$_{th}$, which reaches peak at around 60 or 80 timeslots for different methods. Since κ is a linear sum of ω and χ; and χ keeps decreasing yet ω undergoes an increase period while AoI$_{th}$ is getting larger. We can conclude that a policy that optimizes threshold AoI cannot optimize episodic AoI and AoI threshold violation ratio at the same time, as the indicated trade-off.

3) Impact of Generated Data Amount: We fixed $U = 3$ and AoI$_{th} = 100$ timeslots to investigate the impact of generated data amount Δ at each timeslot. From Fig. 4 and Fig. 5, we see that DRL-UCS(AoI$_{th}$) consistently outperforms nine baselines in term of κ. On average, DRL-UCS(AoI$_{th}$) significantly lower down κ by 26%, 37%, 32%, 43%, 56%, 54%, 51%, 64% and 76% than HATRPO, IMPALA, VDN, QMIX, DRL-freshMCS, DDPG-ESWA, HGS, ES and Random approaches, respectively.

We also observe that κ increases and ζ drops significantly when more data is generated from Fig. 11 and Fig. 12. This is because UAVs need more time to collect one single piece of data and carefully plan their trajectory during less movement time. Thus, with the help of temporal modeling by GTrXL, DRL-UCS(AoI$_{th}$) still achieves lowest $\kappa = 36.16$ timeslots when $\Delta = 1.4$Mb in Beijing, 62% and 54% lower than DRL-freshMCS and DDPG-EWSA, respectively. On the other hand, fewer data amount simplifies the whole task since UAVs can collect more data without moving far away from one spot. If there is no effective exploration mechanism, DRL-based solutions falls into local optima and neglect the unvisited PoIs. Benefited from intrinsic reward, DRL-UCS(AoI$_{th}$) successfully maintains $\kappa = 20.74$ timeslots in San Francisco when generated data amount $\Delta = 0.6$Mb, which is 40% and 51% lower than that of HATRPO and IMPALA, respectively.

D. Trajectory Visualization

In Fig. 13, we show the trajectory of DRL-UCS(AoI$_{th}$) and the best baseline HATRPO when 3 UAVs are deployed. From the trajectories of our approach, we see that each UAV is mainly responsible for a part of workzone. This phenomenon should be attributed to our regional diversity reward given by intrinsic reward. Also, with the help of spatial exploration in intrinsic reward, UAVs successfully cover the entire workzone and find different trajectories to collect data from different PoIs. Most of the time, UAV tends to corporately fly around the area where PoIs are densely distributed to get a lower overall AoI (i.e., cooperative collection in the green area for pursuing data freshness, alone collection in the purple area for maintaining AoI under threshold) and access remote PoIs only when they are nearly going over the AoI$_{th}$ (i.e., black area). This is brought by GTrXL, which models the temporal features to fully consider the generation time of each data and AoI at each PoI in the current timeslot. We also show threshold AoI curves for PoIs located within the purple region on the left side of Fig. 13. Due to the limited mobility and data collection



(a) DRL-UCS(AoI_{th}) (Beijing)

(b) HATRPO (Beijing)

	Area	κ	χ
DRL-UCS(AoI_{th})	🟩	17.65	0.000
	🟪	26.74	0.006
	⬛	36.58	0.052
HATRPO	🟩	21.89	0.008
	🟪	31.67	0.021
	⬛	113.79	0.583

(c) DRL-UCS(AoI_{th}) (San Francisco)

(d) HATRPO (San Francisco)

	Area	κ	χ
DRL-UCS(AoI_{th})	🟩	19.27	0.000
	🟪	27.52	0.009
	⬛	34.93	0.038
HATRPO	🟩	41.27	0.085
	🟪	38.78	0.062
	⬛	139.05	0.583

Fig. 13. Illustrative UAV trajectories in two datasets.

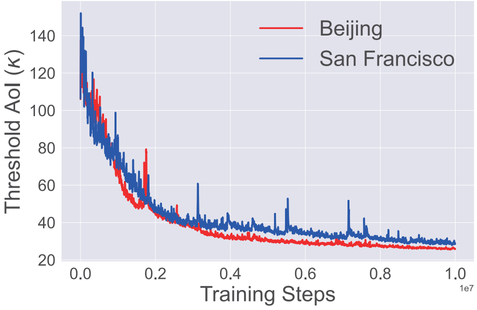


Fig. 14. Convergence curves.

capabilities when deploying 3 UAVs, occasional PoIs may not meet their AoI requirements. However, DRL-UCS(AoI_{th}) guides UAVs to move around to ensure that the majority of PoIs satisfy their AoI requirements thus minimizing the overall κ. Since only limited spatial observations are available to each UAV during decision-making, a temporal modeling mechanism is required for a UAV to remember which area demands immediate data collection service and those areas which can be deferred at the later stage. Meanwhile, different PoIs have diverse prior data generation and collection patterns. Thus these challenges place higher needs on precise temporal modeling and is the key reason we opt to introduce GTrXL for trajectory planning.

Therefore, UAVs are able to balance when to visit which PoI and achieve a lower κ in AoI threshold aware tasks. As a comparison, the best baseline HATRPO suffers from insufficient exploration problems and neglects several PoIs to collect data. Further, without temporal modeling by GTrXL, HATRPO failed to plan UAV trajectories well as there exist lots of redundant overlapping movements.

E. Complexity Analysis

Since DRL-UCS(AoI_{th}) follows the DTDE training paradigm and utilizes IMPALA as the start point of design, it belongs to the set of actor-critic algorithms for fully decentralized MADRL problems. We also show the convergence curve in Fig. 14 for both Beijing and San Francisco datasets. DRL-UCS(AoI_{th}) successfully converges and stabilizes when consuming around 41.6k episodic samples. We show the computational complexity of 7 DRL methods in Table V, as statistics on time cost of making an action in each timeslot t. Although intrinsic reward and GTrXL bring more extra computational cost, we observe that our DRL-UCS(AoI_{th}) runs only negligibly longer than HATRPO and IMPALA in both datasets. Since intrinsic reward is only used in training under DTDE framework and the computational process in GTrXL can be paralleled. Hence our performance is not greatly affected and time cost is still within the scale of millisecond, which is acceptable in practical UAV operations.

TABLE V

Computational Complexity by Time Cost (ms)

Method	Beijing	San Francisco
DRL-UCS(AoI_{th})	3.414	3.917
HATRPO	2.109	2.146
IMPALA	2.717	3.080
VDN	1.982	2.048
QMIX	1.927	1.951
DRL-freshMCS	6.822	7.978
DDPG-EWSA	3.623	3.991

VI. Conclusion

In this paper, we propose DRL-UCS(AoI_{th}), a decentralized MADRL framework for AoI threshold aware UAV crowdsensing. It consists of GTrXL for temporal modeling and intrinsic reward for spatial exploration. Extensive results on Beijing and San Francisco datasets show that DRL-UCS(AoI_{th}) successfully minimizes the AoI and maintains the satisfactory level of AoI threshold, while varying the number of UAVs, AoI threshold and generated data amount in a timeslot.

References

[1] A. T. Campbell et al., "The rise of people-centric sensing," *IEEE Internet Comput.*, vol. 12, no. 4, pp. 12–21, Jul. 2008.

[2] H. Wang et al., "Energy-efficient 3D vehicular crowdsourcing for disaster response by distributed deep reinforcement learning," in *Proc. ACM SIGKDD*, 2021, pp. 3679–3687.

[3] S. Lin, J. Zhang, and L. Ying, "Crowdsensing for spectrum discovery: A waze-inspired design via smartphone sensing," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 750–763, Apr. 2020.

[4] Y. Liu and M. Liu, "An online learning approach to improving the quality of crowd-sourcing," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2166–2179, Aug. 2017.

[5] C. Yi, Y. Chuang, and C. Nian, "Toward crowdsourcing-based road pavement monitoring by mobile sensing technologies," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1905–1917, Aug. 2015.

[6] F. Malandrino, C.-F. Chiasserini, C. Casetti, L. Chiaraviglio, and A. Senacheribbe, "Planning UAV activities for efficient user coverage in disaster areas," *Ad Hoc Netw.*, vol. 89, pp. 177–185, Jun. 2019.

[7] H. Shakhatreh, A. Khreishah, and B. Ji, "UAVs to the rescue: Prolonging the lifetime of wireless devices under disaster situations," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 4, pp. 942–954, Dec. 2019.

[8] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *Proc. IEEE SECON*, Jun. 2011, pp. 350–358.

[9] W. Xu et al., "Minimizing the deployment cost of UAVs for delay-sensitive data collection in IoT networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 2, pp. 812–825, Apr. 2022.

[10] B. Eksioglu, A. V. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Comput. Ind. Eng.*, vol. 57, no. 4, pp. 1472–1483, Nov. 2009.

[11] T. Vidal, "Hybrid genetic search for the CVRP: Open-source implementation and SWAP neighborhood," *Comput. Oper. Res.*, vol. 140, Apr. 2022, Art. no. 105643.

[12] D. Silver et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[13] C Berner et al., "Dota 2 with large scale deep reinforcement learning," 2019, *arXiv:1912.06680*.

[14] C. H. Liu, X. Ma, X. Gao, and J. Tang, "Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1274–1285, Jun. 2020.

[15] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[16] C. H. Liu, Z. Dai, Y. Zhao, J. Crowcroft, D. Wu, and K. K. Leung, "Distributed and energy-efficient mobile crowdsensing with charging stations by deep reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 20, no. 1, pp. 130–146, Jan. 2021.

[17] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: A survey," *Artif. Intell. Rev.*, vol. 55, no. 2, pp. 895–943, Feb. 2022.

[18] W. Wang, Y. Liu, R. Srikant, and L. Ying, "3M-RL: Multi-resolution, multi-agent, mean-field reinforcement learning for autonomous UAV routing," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8985–8996, Jul. 2022.

[19] D. Liu, Z. Du, X. Liu, H. Luan, Y. Xu, and Y. Xu, "Task-based network reconfiguration in distributed UAV swarms: A bilateral matching approach," *IEEE/ACM Trans. Netw.*, vol. 30, no. 6, pp. 2688–2700, Dec. 2022.

[20] W. Xu et al., "Throughput maximization of UAV networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 2, pp. 881–895, Apr. 2022.

[21] C. Rottondi, F. Malandrino, A. Bianco, C. F. Chiasserini, and I. Stavrakakis, "Scheduling of emergency tasks for multiservice UAVs in post-disaster scenarios," *Comput. Netw.*, vol. 184, Jan. 2021, Art. no. 107644.

[22] A. Sawalmeh, N. S. Othman, G. Liu, A. Khreishah, A. Alenezi, and A. Alanazi, "Power-efficient wireless coverage using minimum number of UAVs," *Sensors*, vol. 22, no. 1, p. 223, Dec. 2021.

[23] C. H. Liu, Z. Chen, and Y. Zhan, "Energy-efficient distributed mobile crowd sensing: A deep learning approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1262–1276, Jun. 2019.

[24] C. Luo, M. N. Satpute, D. Li, Y. Wang, W. Chen, and W. Wu, "Fine-grained trajectory optimization of multiple UAVs for efficient data gathering from WSNs," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 162–175, Feb. 2021.

[25] Y. Wang et al., "Task offloading for post-disaster rescue in unmanned aerial vehicles networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 4, pp. 1525–1539, Aug. 2022.

[26] X. Zhong, Y. Guo, N. Li, and Y. Chen, "Joint optimization of relay deployment, channel allocation, and relay assignment for UAVs-aided D2D networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 804–817, Apr. 2020.

[27] Z. Yuan, B. Li, and J. Liu, "Can we improve information freshness with predictions in mobile crowd-learning?" in *Proc. IEEE INFOCOM*, Jul. 2020, pp. 702–709.

[28] Z. Dai et al., "AoI-minimal UAV crowdsensing by model-based graph convolutional reinforcement learning," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2022, pp. 1029–1038.

[29] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proc. ICML*, 2018, pp. 4292–4301.

[30] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. NIPS*, 2017, pp. 6379–6390.

[31] C. Yu et al., "The surprising effectiveness of MAPPO in cooperative, multi-agent games," 2021, *arXiv:2103.01955*.

[32] C. S. de Witt et al., "Is independent learning all you need in the starcraft multi-agent challenge?" 2020, *arXiv:2011.09533*.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[34] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 5872–5881.

[35] Z. Jiang et al., "MDPGT: momentum-based decentralized policy gradient tracking," in *Proc. AAAI*, 2021, pp. 9377–9385.

[36] L. Espeholt et al., "IMPALA: scalable distributed deep-RL with importance weighted actor-learner architectures," in *Proc. ICML*, 2018, pp. 1406–1415.

[37] P. Toth and D. Vigo, "Models, relaxations and exact approaches for the capacitated vehicle routing problem," *Discrete Appl. Math.*, vol. 123, nos. 1–3, pp. 487–512, Nov. 2002.

[38] D. Pisinger and S. Ropke, "A general heuristic for vehicle routing problems," *Comput. Oper. Res.*, vol. 34, no. 8, pp. 2403–2435, Aug. 2007.

[39] J. R. Montoya-Torres, J. López Franco, S. N. Isaza, H. F. Jiménez, and N. Herazo-Padilla, "A literature review on the vehicle routing problem with multiple depots," *Comput. Ind. Eng.*, vol. 79, pp. 115–129, Jan. 2015.

[40] C. K. Joshi, T. Laurent, and X. Bresson, "An efficient graph convolutional network technique for the travelling salesman problem," 2019, *arXiv:1906.01227*.

[41] W. Kool, H. van Hoof, J. Gromicho, and M. Welling, "Deep policy dynamic programming for vehicle routing problems," in *Proc. Int. Conf. Integr. Constraint Program., Artif. Intell., Oper. Res.* Los Angeles, CA, USA: Springer, 2022, pp. 190–213.

[42] X. Chen and Y. Tian, "Learning to perform local rewriting for combinatorial optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[43] P. R. D. O. Costa, J. Rhuggenaath, Y. Zhang, and A. Akcay, "Learning 2-opt heuristics for the traveling salesman problem via deep reinforcement learning," 2020, *arXiv:2004.01608*.

[44] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. Shu, "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5723–5728, May 2020.

[45] X. Liu, Y. Liu, and Y. Chen, "Machine learning empowered trajectory and passive beamforming design in UAV-RIS wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2042–2055, Jul. 2021.

[46] Y. Zhang, Z. Mou, F. Gao, J. Jiang, R. Ding, and Z. Han, "UAV-enabled secure communications by multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11599–11611, Oct. 2020.

[47] N. Zhao, Z. Ye, Y. Pei, Y. Liang, and D. Niyato, "Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 6949–6960, Sep. 2022.

[48] J. Li et al., "Joint optimization on trajectory, altitude, velocity, and link scheduling for minimum mission time in UAV-aided data collection," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1464–1475, Feb. 2020.

[49] B. Bellalta, "IEEE 802.11ax: High-efficiency WLANS," *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 38–46, Feb. 2016.

[50] Y. Zeng, X. Xu, and R. Zhang, "Trajectory design for completion time minimization in UAV-enabled multicasting," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2233–2246, Apr. 2018.

[51] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2334–2360, 3rd Quart., 2019.

[52] E. Parisotto et al., "Stabilizing transformers for reinforcement learning," in *Proc. ICML*, 2020, pp. 7487–7498.

[53] Y. Burda, H. Edwards, A. J. Storkey, and O. Klimov, "Exploration by random network distillation," in *Proc. ICLR*, 2019, pp. 1–17.

[54] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2978–2988.

[55] J. Chung, C. C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[56] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[57] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2011, pp. 316–324.

[58] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. (Feb. 24, 2009). *CRAWDAD Dataset EPFL/Mobility*. [Online]. Available: https://crawdad.org/epfl/mobility/20090224

[59] *DJI Matrice 300 RTK—Built Tough, Works Smart*. Accessed: May 27, 2020. [Online]. Available: https://www.dji.com/cn/matrice-300

[60] J. G. Kuba et al., "Trust region policy optimisation in multi-agent reinforcement learning," in *Proc. ICLR*, 2022, pp. 1–27.

[61] P. Sunehag et al., "Value-decomposition networks for cooperative multi-agent learning," 2017, *arXiv:1706.05296*.

[62] Z. Dai, H. Wang, C. H. Liu, R. Han, J. Tang, and G. Wang, "Mobile crowdsensing for data freshness: A deep reinforcement learning approach," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2021, pp. 1–10.

[63] M. Samir, C. Assi, S. Sharafeddine, D. Ebrahimi, and A. Ghrayeb, "Age of information aware trajectory planning of UAVs in intelligent transportation systems: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12382–12395, Nov. 2020.

[64] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," 2017, *arXiv:1703.03864*.

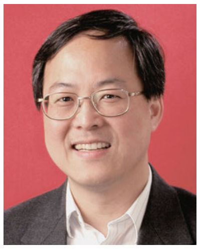
[65] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. ICLR*, 2016, pp. 1–14.

Hao Wang received the B.Eng. degree in software engineering from the Beijing Institute of Technology, China, in 2017, where he is currently pursuing the M.Sc. degree with the School of Computer Science and Technology, under the supervision of Prof. Chi Harold Liu. His current research interests include the problems of mobile crowdsensing and deep reinforcement learning.

Chi Harold Liu (Senior Member, IEEE) received the B.Eng. degree in electronic and information engineering from Tsinghua University, China, in 2006, and the Ph.D. degree in electronic engineering from Imperial College, U.K., in 2010. He has worked with IBM Research, China, and Deutsche Telekom Laboratories. He is currently a Full Professor and the Vice Dean of the School of Computer Science and Technology, Beijing Institute of Technology, China. His current research interests include mobile crowdsensing by deep learning. He is a Fellow of IET and the British Computer Society. He received the IBM First Plateau Invention Achievement Award in 2012, the ACM SigKDD'21 Best Paper Runner-Up Award, and the ACM MobiCom'21 Best Community Paper Runner-Up Award. He serves as the Associate Editor for IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING.

Haoming Yang receives the B.Sc. degree in computer science from the University of California at Berkeley, and the M.Sc. degree in computer science from the University of Southern California, USA. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Beijing Institute of Technology, China, under the supervision of Prof. Chi Harold Liu. His current research interests include the problems of mobile crowdsensing and deep reinforcement learning.

Guoren Wang received the B.Sc., M.Sc., and Ph.D. degrees from the Department of Computer Science, Northeastern University, China, in 1988, 1991, and 1996, respectively. Currently, he is a Full Professor and the Dean of the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. He has published more than 100 research articles. His current research interests include XML data management, query processing and optimization, bioinformatics, high-dimensional indexing, parallel database systems, and cloud data management.



Kin K. Leung (Fellow, IEEE) received the B.S. degree from The Chinese University of Hong Kong in 1980 and the M.S. and Ph.D. degrees from the University of California at Los Angeles, Los Angeles, CA, USA, in 1982 and 1985, respectively. He joined AT&T Bell Labs, NJ, USA, in 1986, and worked at its successors, AT&T Labs and Lucent Technologies Bell Labs, until 2004. Since 2004, he has been the Tanaka Chair Professor with the Electrical and Electronic Engineering (EEE) Department, and the Computing Department, Imperial College, London, U.K. He is currently the Head of the Communications and Signal Processing Group, EEE Department. His current research interests include protocols, optimization, and the modeling of various wireless networks, with applications of novel deep learning techniques. He was a member of Academia Europaea in 2012 and a Fellow of IET in 2021 and the Royal Academy of Engineering in 2022. He received the Distinguished Member of Technical Staff Award from AT&T Bell Labs in 1994. He was a co-recipient of the Lanchester Prize Honorable Mention Award in 1997. He received the Royal Society Wolfson Research Merits Award from 2004 to 2009. He also received several best paper awards and actively served on conference committees and as the journal editors.