AoI-minimal UAV Crowdsensing by Model-based Graph Convolutional Reinforcement Learning

Zipeng Dai
School of Comp. Sci. and Tech.
Beijing Institute of Technology
Beijing, China
3120215520@bit.edu.cn

Chi Harold Liu
School of Comp. Sci. and Tech.
Beijing Institute of Technology
Beijing, China
chiliu@bit.edu.cn

Yuxiao Ye
School of Comp. Sci. and Tech.
Beijing Institute of Technology
Beijing, China
1120181659@bit.edu.cn

Rui Han
School of Comp. Sci. and Tech.
Beijing Institute of Technology
Beijing, China
hanrui@bit.edu.cn

Ye Yuan
School of Comp. Sci. and Tech.
Beijing Institute of Technology
Beijing, China
yuan-ye@bit.edu.cn

Guoren Wang
School of Comp. Sci. and Tech.
Beijing Institute of Technology
Beijing, China
wanggr@bit.edu.cn

Jian Tang
Midea Group
Beijing, China
tangjian22@midea.com

IEEE INFOCOM 2022 - IEEE Conference on Computer Communications | 978-1-6654-5822-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/INFOCOM48880.2022.9796732

***Abstract*—Mobile Crowdsensing (MCS) with smart devices has become an appealing paradigm for urban sensing. With the development of 5G-and-beyond technologies, unmanned aerial vehicles (UAVs) become possible for real-time applications, including wireless coverage, search and even disaster response. In this paper, we consider to use a group of UAVs as aerial base stations (BSs) to move around and collect data from multiple MCS users, forming a UAV crowdsensing campaign (UCS). Our goal is to maximize the collected data, geographical coverage whiling minimizing the age-of-information (AoI) of all mobile users simultaneously, with efficient use of constrained energy reserve. We propose a model-based deep reinforcement learning (DRL) framework called "GCRL-min(AoI)", which mainly consists of a novel model-based Monte Carlo tree search (MCTS) structure based on state-of-the-art approach MCTS (AlphaZero). We further improve it by adding a spatial UAV-user correlation extraction mechanism by a relational graph convolutional network (RGCN), and a next state prediction module to reduce the dependance of experience data. Extensive results and trajectory visualization on three real human mobility datasets in Purdue University, KAIST and NCSU show that GCRL-min(AoI) consistently outperforms five baselines, when varying different number of UAVs and maximum coupling loss in terms of four metrics.**

***Index Terms*—Mobile crowdsensing, Unmanned aerial vehicles, Age of Information, Graph convolutional reinforcement learning.**

I. Introduction

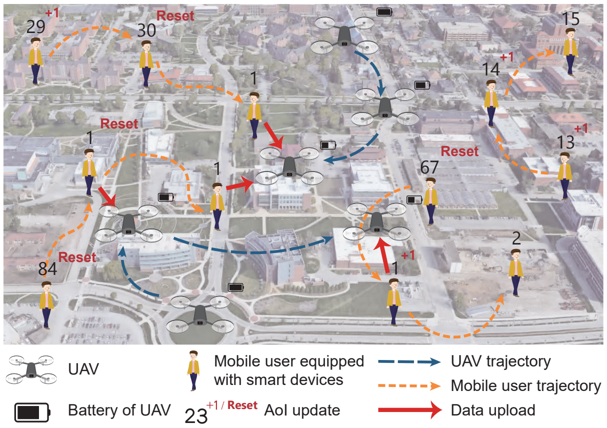


Fig. 1: Overall scenarios for UAV Crowdsensing (UCS).

Mobile crowdsensing (MCS [1]) has been recognized as an efficient and scalable way to acquire data for diverse smart city applications like traffic control and road condition monitoring [2], [3]. Human-centric MCS heavily rely on classical communication facilities like terrestrial base stations (BSs), however they may not be able to cope with spontaneous and temporary mass event like disaster response and public safety, within a tolerable delay. This may become even more severe when the BSs goes out of operation in emergencies. Therefore, we consider to deploy multiple unmanned aerial vehicles (UAVs) as mobile aerial BSs to provide additional network capacity across space and time domains, with the following advantages: (a) UAVs can move more flexibly to provide wireless coverage for underlying mobile users, to form a new UAV crowdsensing (UCS) campaign; (b) UAVs with high-precision control platform, smart sensors and communication interfaces can provide wider coverage for mobile users, to enhance the quality of collected sensory data. Furthermore, delay is critical in such UCS applications to acquire real-time data from mobile users. Thus, we explicitly consider "age of information" (AoI [4]) into our framework as a metric to evaluate the timeliness of data collection, defined by the elapsed period of time after the latest successful transmission of the valid uploaded data from a mobile user to a UAV.

Fig. 1 shows an illustrative example, where UAVs move around and receive data uploaded from multiple mobile users (e.g., students and stuff members in campus environments). The AoI of each mobile user will be reset if his/her collected data is successfully received by a UAV in a timeslot, otherwise it keeps increasing over time. Key challenges include, first, long distance movement to collect user data, saving battery power and keeping data freshness of each user are trading-off for each UAV. Second, mobile users are uncontrollable, thus

978-1-6654-5822-1/22/$31.00 ©2022 IEEE

UAVs need to carefully design their trajectories according to the time-varying spatial distribution of users to achieve our goal. In addition, it is more efficient for UAVs to learn a cooperative control policy and focus on their own responsible areas in a collaborative way.

To calculate the optimal policy, it is hard to formulate the above challenges as a closed-form optimization problem and utilize classical mathematical methods (e.g., Lagrangian multiplier with KKT conditions) to obtain a solution. Furthermore, existing research efforts are paid mainly to form as a Markov Decision Process (MDP) and use value iteration algorithms [5], [6] to find the data transmission scheduling schemes and minimize the average AoI in multi-user uplink systems. However, value iterations cannot deal with large state/action space in a MDP with hundreds of mobile users. Recent achievements [7], [8] along the direction of deep reinforcement learning (DRL) paves the possible way for solving UAV navigation problems in UCS. Most of existing works utilize model-free DRL methods (e.g., DQN [9] and Rainbow [10]) to train a suitable policy directly with experience, and then show the best asymptotic performance by deep neural networks (DNNs). However, these model-free solutions fail to efficiently utilize limited real-world interactions from the task environment. Although we can assume to infinitely explore and exploit from experience data in simulation platforms, it is still impossible to tolerate repeated failures (e.g., obstacle collision, dead battery) in real-world scenarios. As a result, they keep collecting huge amount of training data by trial-and-error, and take long time for the model convergence.

To this end, in this paper, we propose a model-based DRL solution called "GCRL-min(AoI)", using MCTS (AlphaZero) [8] as the start point of design. GCRL-min(AoI) successfully navigates a group of UAVs as mobile aerial BSs to move around and collect data from multiple mobile users. Our contribution is three-fold:

1) We design a novel model-based Monte Carlo tree search (MCTS) structure using state-of-the-art approach MCTS (AlphaZero) as the start point of design, with improvement of adding next state prediction module. It uses less interaction experience than model-free methods, by learning a predictive state transition model to generate possible next states, which reduces excessive dependence on simulation platforms.
2) We employ a relational graph convolutional network (RGCN) to extract spatial correlations between UAVs and mobile users, to retrieve reliable state representations. It helps each UAV to pay attention to different groups of mobile users and learn to collaborate in a large-scale task area.
3) We perform extensive experiments on three real-world human trajectory datasets in campus environments, namely: Purdue University (USA), NCSU (USA) and KAIST (South Korea). Results confirm that GCRL-min(AoI) has robust improvements on data collection ratio and episodic AoI, compared with five other baselines.

The remainder of the paper is organized as follows. Section II reviews the related work. Section III presents system model. Section IV describes problem definition and formulation. Section V gives preliminaries. Section VI describes our proposed approach GCRL-min(AoI). Section VII gives the evaluation results. Finally, Section VIII concludes the paper.

II. Related Work

In MCS, many existing solutions are proposed to utilize UAVs for efficient data collection [11], [12]. For example, Zhou *et al.* in [13] considered energy-efficient task allocation and route planning problems by utilizing fixed-wing UAVs as workers. Liu *et al.* in [14] developed a decentralized DRL framework to maximize the energy efficiency of UAVs, while ensuring UAV coverage and user fairness. Recently, AoI was proposed as a metric to quantify the freshness of time-sensitive information [15]. Hsu *et al.* in [5], [16] proposed to minimize long-run average AoI by designing a dynamic MDP-based transmission scheduling schemes over a wireless broadcast network. Hu *et al.* in [17] designed a joint energy transfer and data collection time allocation problem in UAV-assisted wireless powered IoT system, and then solved it by combining dynamic programming and heuristic algorithms. Li *et al.* in [19] formulated data collection maximization problems to deal with data collection from sensors to UAVs, and then devised approximation algorithms. Liu *et al.* in [20] navigated a group of multi-antenna UAVs to minimize AoI in a fixed sensor network, with constrained energy reserve. However, none of these approaches consider AoI minimization in UCS for multiple mobile users. To the best of our knowledge, we are one of the first along this direction.

III. System Model

In our considered UCS scenario, we define a time-slotted system and divide the time range into T equal timeslots. Let $\mathcal{U} \triangleq \{u|u = 1, 2, \cdots, U\}$ and $\mathcal{M} \triangleq \{m|m = 1, 2, \cdots, M\}$ denote UAVs and mobile users in a 3D target area, respectively. We can convert their GPS coordinates (i.e., longitude, latitude and altitude) to the corresponding position (x_t^u, y_t^u, h^u) or $(x_t^m, y_t^m, 0)$. In addition, there are tall buildings $\mathcal{B} \triangleq \{b|b = 1, 2, \cdots, B\}$ with height h^b, which UAVs should avoid if $h^b \geq h^u$. For simplicity, we assume UAVs and users move around at 2D planes, and note that our approach can be easily extend to 3D navigation scenarios by adding altitude control. In each timeslot $[t, t+1)$, each UAV u spends τ time moving to a certain direction $\vartheta_{\text{move},t}^u \in [0, 2\pi)$ at speed $v_{\text{move},t}^u \in [0, v_{\max}]$, where $v_{\max}$ denotes the maximum speed of a UAV. Meanwhile, each mobile user m moves from $(x_t^m, y_t^m, 0)$ to $(x_{t+1}^m, y_{t+1}^m, 0)$ and collects data individually, given the expected sampling frequency v_{collect}^m of the user equipped smart device.

The system works as follows. At the beginning, U UAVs are deployed at the same origin with full energy reserve $E_{\max}$. Meanwhile, M mobile users start to move around to collect data with initial data volume $\delta_0^m = 0$. In timeslot $[t, t+1)$, each user m collects data and tries to upload all the remaining data to the nearest UAV u (as a mobile aerial BS). Without loss of generality, we consider a mmWave-based data uplink system where mobile users and UAVs are transmitters (Tx) and receivers (Rx), respectively. Note that our system model can work also at sub-6 GHz bands. Following [21], [22], we assume that radio interference does not have a major effect, which is a common assumption for most mmWave-based systems with directional antennas; also the system in this paper is noise-limited. Recall that path loss (PL) models for the line-of-sight (LoS) and non-line-of-sight (NLoS) links at certain mmWave frequencies are given as:

$$\begin{aligned} PL_t^{\mathrm{LoS}}(u,m) &= \alpha^{\mathrm{LoS}} + 10\beta^{\mathrm{LoS}} \log\left(d_t^{\mathrm{3D}}(u,m)\right), \\ PL_t^{\mathrm{NLoS}}(u,m) &= \alpha^{\mathrm{NLoS}} + 10\beta^{\mathrm{NLoS}} \log\left(d_t^{\mathrm{3D}}(u,m)\right), \end{aligned} \tag{1}$$

where α^{LoS}, β^{LoS}, α^{NLoS}, β^{NLoS} are environment parameters on floating intercept and slope, and $d_t^{\mathrm{3D}}(u,m)$ is the 3D distance between mobile user m and UAV u. Obviously, $PL_t(u,m)$ increases as $d_t(u,m)$ increases. For LoS and NLoS, we consider a ground-to-air channel model, where a UAV u locates at height h^u, while each user is modeled as cylinders with the average height h^{user} and the average diameter of g^{user}. For a snapshot analysis, we assume mobile users' density follows Poisson distribution with parameter λ. A mobile user carrying smart device is assumed to located at height h^{device}, where $h^{\mathrm{device}} < h^{\mathrm{user}}$. Hence, if a user uploads data to a UAV, the surrounding buildings and other users may block the LoS transmission path. According to [23], we calculate the probability of LoS in timeslot $[t, t+1)$ by:

$$\mathbb{P}_t^{\mathrm{LoS}}(u,m) = \exp\left(-\lambda g_{\mathrm{user}} d_t^{\mathrm{2D}}(u,m) \frac{h^{\mathrm{user}} - h^{\mathrm{device}}}{h^u - h^{\mathrm{device}}}\right), \tag{2}$$

where $d_t^{\mathrm{2D}}(u,m)$ is the Euclidean distance between user m and UAV u. Thus, their average path loss is denoted by $PL_t(u,m) = \mathbb{P}_t^{\mathrm{LoS}}(u,m) \cdot PL_t^{\mathrm{LoS}}(u,m) + \mathbb{P}_t^{\mathrm{NLoS}}(u,m) \cdot PL_t^{\mathrm{NLoS}}(u,m)$, where $\mathbb{P}_t^{\mathrm{NLoS}}(u,m) = 1 - \mathbb{P}_t^{\mathrm{LoS}}(u,m)$.

Furthermore, due to the minimum acceptable received power level, UAVs at a high altitude have limited sensing range for data collection. According to 5G NR [24], we choose maximum coupling loss (MCL) to represent UAV's maximum sensing range in each timeslot, defined as the maximum loss in the conducted power level that a system can tolerate and still be operational. Let G^{Tx} and G^{Rx} be the gains of Tx and RX antennas, respectively. For each user m, we define data upload and AoI update as follows.

Definition 1. ***(Collected Data Upload)*** *In timeslot $[t, t+1)$, a user m tries to upload all remaining data to the nearest UAV u. The uploading process is successful if $PL_t(u,m)$ is tolerable. Thus, the user m's remaining data δ_t^m is updated to δ_{t+1}^m by:*

$$\delta_{t+1}^m = \begin{cases} 0, & \text{if } PL_t(u,m) - G^{\mathrm{Tx}} - G^{\mathrm{Rx}} \leq MCL \\ \delta_t^m + v_{\mathrm{collect}}^m \cdot \tau, & \text{otherwise.} \end{cases} \tag{3}$$

Considering different data collection capability v_{collect}^m of each user's smart device, the successfully collected data amount is computed by $\sum_{m=1}^{M} T \cdot v_{\mathrm{collect}}^m \cdot \tau - \delta_T^m$.

Definition 2. ***(AoI Update)*** *In timeslot $[t, t+1)$, we use AoI κ_t^m of user m to describe the timeliness of data upload, updated by:*

$$\kappa_{t+1}^m = \begin{cases} 1, & \text{if } PL_t(u,m) - G^{\mathrm{Tx}} - G^{\mathrm{Rx}} \leq MCL \\ \kappa_t^m + 1, & \text{otherwise.} \end{cases} \tag{4}$$

Following [25], we further consider the the limited battery life of rotary-wing UAVs. The propulsion energy consumption of UAV u during each timeslot $[t, t+1)$ is given by $\omega_t^u = \tau \cdot \Big[c_1\Big(1 + \frac{3\cdot(v_{\mathrm{move},t}^u)^2}{(v_{\mathrm{tip}})^2}\Big) + c_2\Big(\sqrt{1 + \frac{(v_{\mathrm{move},t}^u)^4}{4\bar{v}^4}} - \frac{(v_{\mathrm{move},t}^u)^2}{2\bar{v}^2}\Big)^{\frac{1}{2}} + \frac{1}{2}c_3(v_{\mathrm{move},t}^u)^3\Big]$, where c_1, c_2, c_3 are constants that depend on UAV's weight, rotors, blades and air density. v_{tip} and $\bar{v}$ are the tip speed and average speed of the rotor, respectively. If a UAV u runs out of battery, it will stop receiving users' data immediately, as task failure in this paper.

IV. Problem Definition and Formulation

A. Problem Definition

We first define four evaluation metrics to justify the completion of UCS tasks. First is data collection ratio that describes the collected data amount over all the available data of mobile users, denoted by:

$$\psi = \frac{\sum_{m=1}^{M} T \cdot v_{\mathrm{collect}}^m \cdot \tau - \delta_T^m}{\sum_{m=1}^{M} T \cdot v_{\mathrm{collect}}^m \cdot \tau}, \tag{5}$$

Second is the average user coverage $\bar{\rho}$ that evaluates the average covered mobile users over time of all UAVs:

$$\bar{\rho} = \frac{1}{T} \sum\nolimits_{t=1}^{T} |\mathcal{K}_t|, \tag{6}$$

where $\mathcal{K}_t \subset \mathcal{M}$ is the set of mobile users whose data are successfully uploaded to UAVs, i.e., $PL_t(u,m) - G^{\mathrm{Tx}} - G^{\mathrm{Rx}} \leq MCL$. Third is the average energy consumption ratio $\bar{\zeta} = \frac{1}{U \cdot E_{\max}} \sum_{u=1}^{U} \sum_{t=1}^{T} \omega_t^u$. Finally, the episodic AoI is defined as:

$$\bar{\kappa} = \frac{1}{T} \frac{1}{M} \sum\nolimits_{t=1}^{T} \sum\nolimits_{m=1}^{M} \kappa_t^m, \tag{7}$$

Note that $\bar{\kappa} \geq 1$. Our goal is to maximize data collection ratio ψ and average user coverage $\bar{\rho}$ simultaneously, while minimizing episodic AoI $\bar{\kappa}$ given limited UAV energy.

B. Problem Formulation

On designing a DRL model, we formulate the considered problem as a MDP, containing: state space $\mathcal{S} \triangleq \{\boldsymbol{s}_t\}$, action space $\mathcal{A} \triangleq \{\boldsymbol{a}_t\}$, state transition model $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, and reward function $r(\boldsymbol{s}_t, \boldsymbol{a}_t)$ in each timeslot $[t, t+1)$.

A state $\boldsymbol{s}_t$ includes two groups of tensors. The first is the current 3D position (x_t^u, y_t^u, h^u) of each UAV u, and remaining energy $E_{\max} - \sum_{i=1}^{t-1} \omega_i^u$; and the second is each mobile user m's position (x_t^m, y_t^m), remaining data δ_t^m and AoI κ_t^m at the beginning of timeslot $[t, t+1)$. $\boldsymbol{a}_t$ denotes a joint decision, including each UAV's action $\boldsymbol{a}_t^u = (\vartheta_{\mathrm{move},t}^u, v_{\mathrm{move},t}^u)$. $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the state-transition function from $\boldsymbol{s}_t$ to $\boldsymbol{s}_{t+1}$, according to the system model in Section III. Without loss of generality, we assume that UAVs observe $\boldsymbol{s}_t$ at the beginning of each timeslot $[t, t+1)$, and then take actions $\boldsymbol{a}_t$. Next, the state transits to $\boldsymbol{s}_{t+1}$ by model $\mathcal{T}$, and finally we get a certain reward $r_t = r_t^+ + r_t^-$ at the end of this timeslot to evaluate this decision, where:

$$r_t^+ = \frac{1}{M}\sum_{m=1}^{M} c_{\text{collect}}(\delta^m - \delta^m_{t+1}) + c_{\text{AoI}}(\kappa^m_t - \kappa^m_{t+1}). \quad (8)$$

Here, r_t^- denotes the penalty when a UAV u hits obstacles or runs out of energy; $c_{\text{collect}}, c_{\text{AoI}}$ are constants. We aim to train a policy π that defines a probability distribution over $\boldsymbol{a}_t$ for each state $\boldsymbol{s}_t$. The objective of π is to select the action to maximize the total sum of discounted future reward with a discount factor $\gamma \in [0, 1)$, as:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}\big[\textstyle\sum_{t=1}^{T} \gamma^t\, r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)\big] \\ \text{s.t.} \quad & \pi(\boldsymbol{s}_t) \in \mathcal{A}, \quad \mathcal{T}(\boldsymbol{s}_t, \boldsymbol{a}_t) \in \mathcal{S}. \end{aligned} \quad (9)$$

Formally, Eqn. (9) is NP-hard, and thus intractable. It is hard to form it as a traditional constrained optimization problem, since the policy π cannot be explicitly expressed in a closed-form equation. Alternatively, we consider utilizing DRL method.

V. Preliminaries

The state-of-the-art tree-search method MCTS (AlphaZero) [8] estimates the value of each nodes (to save recursive state value on a tree) by utilizing value prediction networks [26], rather than maintaining a complex value function by exhaustive search. The state value estimation process of an N-depth MCTS (AlphaZero) is denoted by:

$$V^{(n)}(\boldsymbol{s}_t) = \begin{cases} f^{\text{val}}(\boldsymbol{s}_t), \text{ if} & n = 1 \\ \frac{1}{n}V^{(1)}(\boldsymbol{s}_t) + \frac{n-1}{n}\max_{\boldsymbol{a}}\Big(& \\ \quad r(\boldsymbol{s}_t, \boldsymbol{a}) + \gamma V^{(n-1)}(\boldsymbol{s}_{t+1})\Big) & , \text{ otherwise.} \end{cases} \quad (10)$$

where f^{val} is a DNN that estimates the certain value of given state $\boldsymbol{s}_t$; $V^{(n)}(\boldsymbol{s}_t)$ is the node value of $\boldsymbol{s}_t$ on the n-th layer of the tree, updated layer by layer recursively. The learning schemes replaces the traditional exhaustive search by just looking ahead N steps, which has been used for decision-making in complex search problems, from the game of Go [7], [8] to robot navigation [27]. Although model-free DRL approaches have achieved a great success in some areas, severe and costly failures (e.g., obstacle collision, dead battery) are still big problem for UAVs in real world scenarios.

By contrast, model-based DRL methods increase the sample data efficiency, with fewer interactions to real environment. Sutton *et al.* proposed Dyna-Q algorithm [28] in which the construction of the environment model and sampling are carried out simultaneously. The constructed environment model can decrease the interaction between the agent and the environment and make the agent perform value iterations quickly, which motivates us to consider a model-based solution.

VI. Proposed Solution: GCRL-min(AoI)

GCRL-min(AoI) mainly consists of a novel model-based MCTS structure using approach MCTS (AlphaZero) as the start point of design, with improvement introduced as follows.

A. UAV-User Spatial Correlation Extraction by Relational Graph Convolutional Network

Understanding the interactions (i.e., spatial relations) between UAVs and mobile users is key to efficiently navigate multiples UAVs to fly around as mobile aerial BSs. Previous works [30] show that using spatial self-attention or graph neural network (GNN) can improve both interpretation and performance of collision avoidance, by modeling one-way human-robot interactions. Although our problem is far more difficult than simply avoiding obstacles, it inspires and motivates us to map UAVs and mobile users as "nodes" in a directed graph, denoted as $\mathcal{G}_t = (\mathcal{N}, \mathcal{E})$, where $|\mathcal{N}| = U + M$. Here $\mathcal{N}$ and $\mathcal{E}$ are sets of nodes and edges in $\mathcal{G}_t$, respectively. The edge $e_{i,j} \in \mathcal{E}$ indicates how much attention a node i pays to a node j, or the importance of a node j to a node i. Since mobile users' intentions or hidden policies of movement are not known as a priori, this pairwise relation is also not known. However, it can be inferred with a pairwise similarity function (as relation inference). After the relations between all nodes are inferred, we utilize RGCN [31] to propagate relational information from node to node, and compute the state representations for each UAV and user. As shown in Fig. 2, the forward process is depicted by:

$$\boldsymbol{z}_t^u = f^{\text{UAV}}(\boldsymbol{s}_t^u), \quad \forall u \in \mathcal{U}, \quad (11a)$$

$$\boldsymbol{z}_t^m = f^{\text{user}}(\boldsymbol{s}_t^m), \quad \forall m \in \mathcal{M}, \quad (11b)$$

$$\mathbf{Z}_t = \text{concat}\Big(\{\boldsymbol{z}_t^u\}_{u=1}^U,\ \{\boldsymbol{z}_t^m\}_{m=1}^M\Big), \quad (11c)$$

$$\mathbf{C}_t = \Big[f(\boldsymbol{z}_t^i, \boldsymbol{z}_t^j)\Big]_{|\mathcal{N}|\times|\mathcal{N}|} = \text{softmax}(\mathbf{Z}_t\mathbf{W}_c\mathbf{Z}_t^\top), \quad (11d)$$

$$\begin{aligned} &\mathbf{H}_t^{(0)} = \mathbf{Z}_t, \\ &\mathbf{H}_t^{(l+1)} = \text{ReLU}(\mathbf{C}_t\mathbf{H}_t^{(l)}\mathbf{W}_h^{(l)}) + \mathbf{H}_t^{(l)}, \end{aligned} \quad (11e)$$

$$\{\boldsymbol{\eta}_t^u\}_{u=1}^U,\ \{\boldsymbol{\eta}_t^m\}_{m=1}^M = \text{split}\Big(\mathbf{H}_t^{(-1)}\Big) \quad (11f)$$

First, we initialize values of vertices $\mathcal{N}$ by state $\boldsymbol{s}_t$ in timeslot $[t, t+1)$. Since each UAV's part of state $\boldsymbol{s}_t^u = \{x_t^u, y_t^u, h_t^u, E_{\max} - \sum_{i=1}^{t-1}\omega_i^u\}$ and each user part of state $\boldsymbol{s}_t^m = \{x_t^m, y_t^m, \delta_t^m, \kappa_t^m\}$ have different meanings and scales, we use two multi-layer perceptions (MLPs) $f^{\text{UAV}}(\cdot)$ and $f^{\text{user}}(\cdot)$ to embed them into a latent space with the same dimension (see Eqn. (11a)–(11b)). Then, we concatenate all these features to a matrix $\mathbf{Z}_t$ as GCN inputs (see Eqn. (11c)). As discussed in Section VII-C, we finally use an embedded Gaussian [32] as the similarity function to compute a UAV-user relational matrix $\mathbf{C}_t$. The pairwise form of $c_t^{ij} \in \mathbf{C}_t$ is given by $f(\boldsymbol{z}_t^i, \boldsymbol{z}_t^j) = \exp[(\mathbf{W}_i\boldsymbol{z}_t^i)^\top(\mathbf{W}_j\boldsymbol{z}_t^j)]$, while the matrix form of $\mathbf{C}_t$ is given by $\mathbf{C}_t = \text{softmax}(\mathbf{Z}_t\mathbf{W}_c\mathbf{Z}_t^\top)$, where $\mathbf{W}_c = \mathbf{W}_i\mathbf{W}_j^\top$ (see Eqn. (11d)). A learned correlation is illustrated in Fig. 2, where the thickness of the line indicates the strength of correlations between UAVs and users.

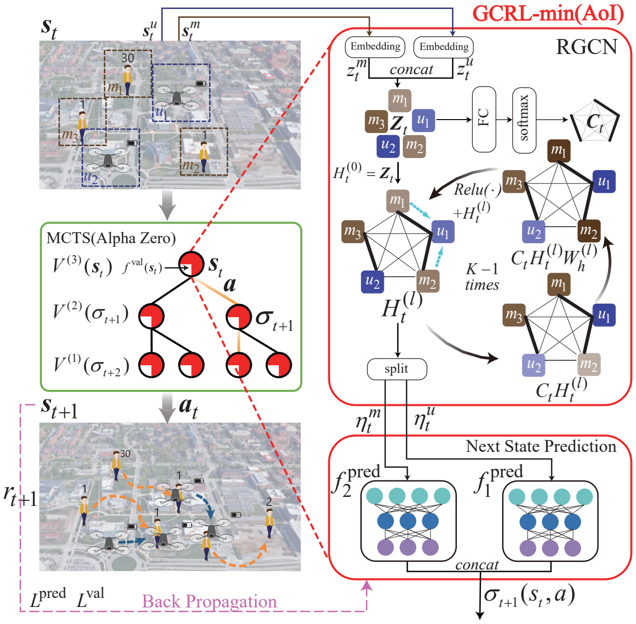


Fig. 2: Diagram of GCRL-min(AoI).

Algorithm 1 Pseudocode of GCRL-min(AoI)

```
Initialize parameters of $f^{\text{UAV}}, f^{\text{user}}, f_1^{\text{pred}}, f_2^{\text{pred}}, f^{\text{val}}, \mathcal{G}$;
for Episode in $1, 2, \cdots$ do
    for Timeslot $[t, t+1)$ in $1, 2, \cdots, T$ do
        Get the current state $\boldsymbol{s}_t$;
        Calculate the state representation $\{\boldsymbol{\eta}_t^u\}_{u=1}^U$, $\{\boldsymbol{\eta}_t^m\}_{m=1}^M$ of $\boldsymbol{s}_t$ with $f^{\text{UAV}}, f^{\text{user}}$, respectively;
        for $\boldsymbol{a}$ in action space $\mathcal{A}$ do
            Predict $\boldsymbol{\sigma}_{t+1}(\boldsymbol{s}_t, \boldsymbol{a})$ with $f_1^{\text{pred}}, f_2^{\text{pred}}$;
        end for
        Estimate state value $V^{(N)}(\boldsymbol{s}_t)$ by Eqn. (13);
        Choose the best action as $\boldsymbol{a}_t$ by Eqn. (14);
        Interact by $\boldsymbol{a}_t$, and get $r_t, \boldsymbol{s}_{t+1}$;
        Compute $L^{\text{predict}}$, $L^{\text{value}}$ by Eqn. (15) and (16);
        Update $f_1^{\text{pred}}, f_2^{\text{pred}}$ by minimizing $L^{\text{pred}}$;
        Update $f^{\text{val}}$ by minimizing $L^{\text{val}}$;
        $t = t + 1$;
    end for
end for
```

Then, in each timeslot $[t, t+1)$, with the feature matrix $\mathbf{Z}_t$ and relation matrix $\mathbf{C}_t$, we use a RGCN to compute the pairwise interaction features. The message passing rule is defined by Eqn. (11e), where $\mathbf{W}_h^{(l)}$ is a layer-specific trainable weight matrix, and $\mathbf{H}_t^{(l)}$ is the node-level features at level l weighted by the UAV-user relations stored in matrix $\mathbf{C}_t$. Let $\mathbf{H}_t^{(0)} = \mathbf{Z}_t$. After multiple message passing processes by layer propagation, we split the last-level features $\mathbf{H}_t^{(-1)}$ into UAV part of features $\{\boldsymbol{\eta}_t^u\}_{u=1}^U$, and user part of features $\{\boldsymbol{\eta}_t^m\}_{m=1}^M$. Note that the above process learns the spatial correlation between UAVs and users without changing the dimension of features (i.e., $\boldsymbol{\eta}$ and $\boldsymbol{z}$ have the same dimensions), which is beneficial for user location prediction and state value estimation in the following.

B. Model-based MCTS (AlphaZero) with Improvement of Next State Prediction for Multi-UAV Navigation

As shown in Fig. 2, we utilize MCTS (AlphaZero) as an N-step planning module for multiple UAVs as the start point of the design, performing state value estimation up to N steps in the future, and select the action with the maximum N-step return, denoted by $\boldsymbol{a}_t = \max_{\boldsymbol{a}} \left(r(\boldsymbol{s}_t, \boldsymbol{a}) + \gamma V^{(N)}(\boldsymbol{s}_{t+1})\right)$. Here $V^{(N)}$ is the value of head node in MCTS (AlphaZero), defined by Eqn. (10). However, MCTS (AlphaZero) estimates state values based on many simulated trajectories $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$. As said earlier, a practical UCS system in real-world scenarios should be online learning. Therefore, inspired by several model-based solutions [28], [29], we propose a novel model-based MCTS structure for multi-UAV navigation, which learns policy π and transition model $\mathcal{T}$ simultaneously. In other words, it can not only optimize policy π by estimating state value V, but also generates the estimated next state $\boldsymbol{\sigma}_{t+1}$ rather than directly getting $\boldsymbol{s}_{t+1}$ from simulations.

1) Next state prediction: With the help of RGCN, we use each UAV u's features $\boldsymbol{\eta}_t^u$ to predict its next state (including its location and remaining energy level) by a shared Multilayer Perceptron (MLP) f_1^{pred}, given a certain action $\boldsymbol{a}^u$ from $\mathcal{A}$. Similarly, considering user motion is not influenced by each UAV's action, we directly use $\boldsymbol{\eta}_t^m$ to predict a user m's next state (including its next location, remaining data and AoI) by another shared MLP f_2^{pred}. We concatenate these estimated states and generate the estimated next state $\boldsymbol{\sigma}_{t+1}$, as:

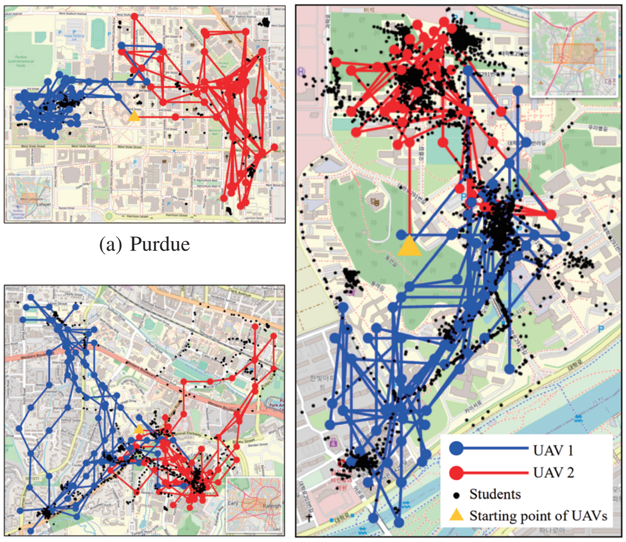
$$\boldsymbol{\sigma}_{t+1}(\boldsymbol{s}_t, \boldsymbol{a}) = \text{concat}\Big(\{f_1^{\text{pred}}(\boldsymbol{\eta}_t^u, \boldsymbol{a}^u)\}_{u=1}^U, \{f_2^{\text{pred}}(\boldsymbol{\eta}_t^m)\}_{m=1}^M\Big). \tag{12}$$

Note that $\boldsymbol{\sigma}_{t+1}(\boldsymbol{s}_t, \boldsymbol{a})$ has the same dimensions as $\boldsymbol{s}_{t+1}$, that represents the predicted next state by taking action $\boldsymbol{a}$, given the observed state $\boldsymbol{s}_t$. Here we set two shared MLPs $f_1^{\text{pred}}, f_2^{\text{pred}}$ instead of a powerful MLP with all UAVs' and users' $\boldsymbol{\eta}_t$ as inputs, in order to scale well with any amount of UAVs/users, given that some users or even UAVs may quit the task sometime in between.

2) State value estimation: Imperfect learned value functions can lead to suboptimal actions due to local minima. With the help of MLP f^{val}, N-depth MCTS (AlphaZero) looks forward N steps to the future to provide a better estimate of the state values, which is beneficial to find a far-sighted policy π and save time of doing exhaustive search like the traditional tree search methods. As shown in Fig. 2, our value estimation module can predict the N-step value $V^{(N)(\boldsymbol{s}_t)}$ of the state, by simply modifying Eqn. (10) as:

$$V^{(n)}(\boldsymbol{s}_t) = \begin{cases} f^{\text{val}}(\boldsymbol{s}_t), \text{ if } n = 1 \\ \dfrac{1}{n} V^{(1)}(\boldsymbol{s}_t) + \dfrac{n-1}{n} \max\limits_{\boldsymbol{a}} \Big(\\ \quad r(\boldsymbol{s}_t, \boldsymbol{a}) + \gamma V^{(n-1)}(\boldsymbol{\sigma}_{t+1}) \Big) \end{cases}, \text{ otherwise.} \tag{13}$$

where $\boldsymbol{\sigma}_{t+1}$ is calculated by Eqn. (12). Similar to MCTS (AlphaZero), we choose the best action as $\boldsymbol{a}_t$ in timeslot



(a) Purdue

(b) NCSU

(c) KAIST

Fig. 3: Illustrative UAV trajectories in three datasets.

$[t, t+1)$ by:

$$\boldsymbol{a}_t = \max_{\boldsymbol{a}} \Big(r(\boldsymbol{s}_t, \boldsymbol{a}) + \gamma V^{(N)}(\boldsymbol{\sigma}_{t+1}) \Big). \tag{14}$$

C. Algorithm Details and Computational Complexity Analysis

Pseudocode of GCRL-min(AoI) is given in Algorithm 1. First, we adopt Xavier uniform initializer [33] for DNN parameters (Line 1). Then, given state $\boldsymbol{s}_t$ in each timeslot $[t, t+1)$, we calculate the state representation of UAVs and users by RGCN (Line 3-5). After predicting the next state $\boldsymbol{\sigma}_{t+1}(\boldsymbol{s}_t, \boldsymbol{a})$ of executing available actions $\boldsymbol{a} \in \mathcal{A}$ (Line 6-8), we calculate the N-step state value $V^{(N)}(\boldsymbol{s}_t)$, and choose the best actions $\boldsymbol{a}$ for all UAVs as $\boldsymbol{a}_t$ (Line 9-10). Next, we use $\boldsymbol{a}_t$ to interact with the environment and get the reward r_t and real next state $\boldsymbol{s}_{t+1}$ (Line 10). Finally, we calculate the state prediction loss L^{pred} and value estimation loss L^{val} by:

$$L^{\text{pred}} = \Big[\boldsymbol{s}_{t+1} - \boldsymbol{\sigma}_{t+1}(\boldsymbol{s}_t, \boldsymbol{a}_t) \Big]^2, \tag{15}$$

$$L^{\text{val}} = \Big[r_t + \gamma V^{(N)}(\boldsymbol{s}_{t+1}) - V^{(N)}(\boldsymbol{s}_t) \Big]^2. \tag{16}$$

We optimize parameters of the prediction network f_1^{pred}, f_2^{pred} and value network f^{val}, by minimizing L^{pred} and L^{val}, respectively.

GCRL-min(AoI) is based on a tree search architecture with several fully connected (FC) layers, whose computational complexity can easily be optimized by some existing methods [34], [35] as:

$$O\Big(\frac{\chi \cdot D^{\text{child}} \cdot I \cdot \sum_{i=1}^{\Omega} D^{\text{in}} D^{\text{out}}}{\Lambda} \Big), \tag{17}$$

where χ, D^{child}, I, Λ denote the number of parallel searches, children per node, iterations of recursion and available CPU cores respectively; Ω is the number of FC layers, and D^{in}, D^{out} are the dimensions of input vector and output vector of the i-th FC layers. Note that $D^{\text{child}} \leq |\mathcal{A}|$ in our UCS problem.

VII. Performance Evaluation

A. Dataset Descriptions and Simulation Settings

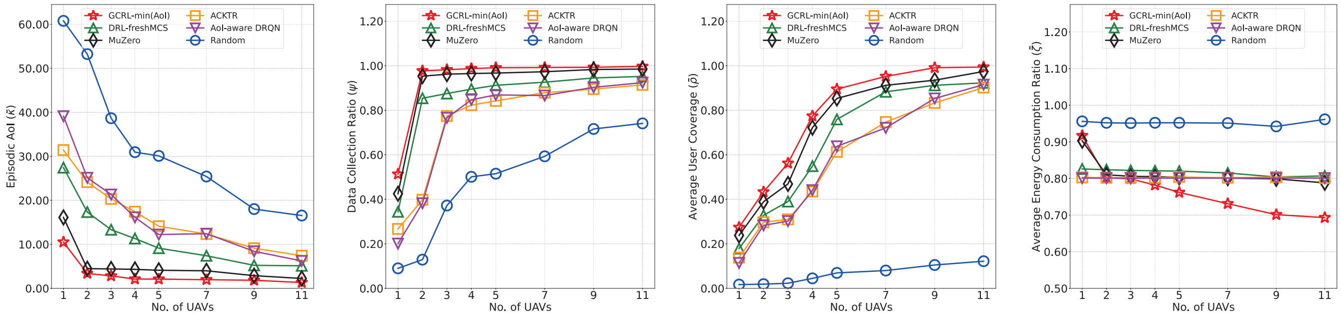
We use three real-world student trajectory data from Purdue University [36], NCSU and KAIST [37]. The traces are generated by students who lived in campus dormitories carrying smartphones with GPS receivers. Google Map is used to mark the campus map data, including student positions, shapes of buildings, lakes and mountains in the environment. Note that there exists lots of GPS data offsets, which are removed by setting the maximum tolerated error 100m for data preprocessing, and eventually we have 59, 33, 92 traces from Purdue University, NCSU and KAIST datasets, respectively. Each trace corresponds to a mobile user.

In our simulation, by referring to the technical report of industrial UAVs like DJI Matrice 600 [38], we consider $T = 120$ with $\tau = 15$ seconds and the battery capacity is $E_{\max} = 4500$mAh. Following [25], we set $c_1 = 79.8563$, $c_2 = 88.6279$, $c_3 = 0.0185$, $v_{\text{tip}} = 120$m/s, $\bar{v} = 4.03$m/s and $v_{\max} = 18$m/s for calculating energy consumption of a UAV. Following [23], we investigate mmWare bands at 28GHz and set $h^u = 120$m, $h^{\text{user}} = 1.7$m, $h^{\text{device}} = 1.3$m, $g^{\text{user}} = 0.5$m, $\lambda = 0.005$, $g^{\text{Tx}} = 0$dB $g^{\text{Rx}} = 5$dB for calculating the probability of LoS by Eqn. (2). We set $\alpha^{\text{LoS}} = 84.64$dB, $\alpha^{\text{NLoS}} = 113.63$dB, $\beta^{\text{LoS}} = 1.55$ and $\beta^{\text{NLoS}} = 1.16$ for calculating path loss, by Eqn. (1). Note that, UAVs should avoid any tall building where $h^b \geq 120$m. For model training, we set $\gamma = 0.95$, $c_{\text{collect}} : c_{\text{AoI}} = 1 : 10$, and learning rate, batch size as 0.001 and 128, respectively.

We use Pytorch 1.8.1 to implement our proposed solution, and all the codes are run on Ubuntu 18.04.2 LTS with Intel(R) Xeon(R) Gold 6238 CPU @2.10GHz with 112 CPU cores. In each following experiment, we train 500 episodes and select the model with lowest episodic AoI for testing. We conduct four sets of experiments, including impact of hyperparameters, time and space cost comparison, ablation study and comparing with five baselines. We utilize episodic AoI $\bar{\kappa}$, data collection ratio ψ, average user coverage $\bar{\rho}$, and average energy consumption ratio $\bar{\zeta}$ as four metrics for comparisons.

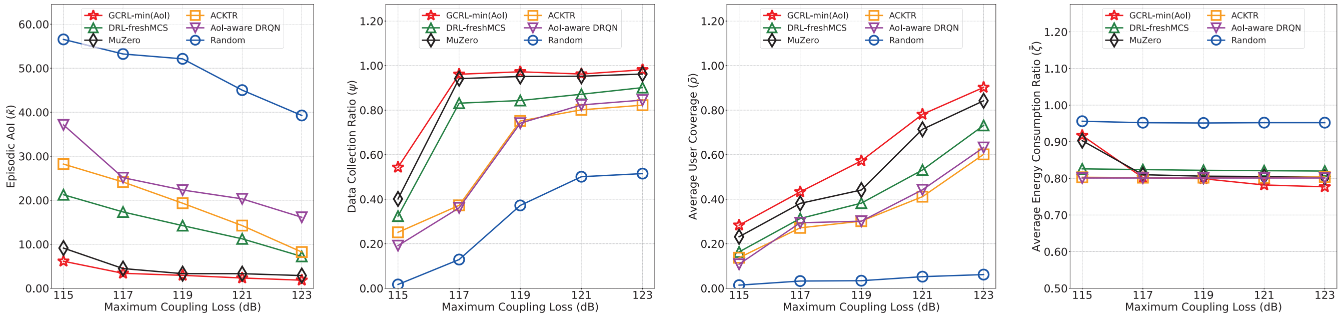
B. Illustrative Student/UAV Trajectories

In Fig. 3, we show the movement trace of UAVs/students when $U = 2$ and find noticeable cooperation among UAVs. Each UAV is mainly responsible for a part of the task area and always moves back and forth. This is because that (a) since each UAV has limited coverage while students keep moving and collecting data, one single shot of sensing is obviously not enough to collect all remaining data, (b) in order to optimize episodic AoI, UAVs should fully consider each student's AoI, thus sometimes flying around the corners to access remote students, as shown in Fig. 3b, and (c) a good cooperative strategy can decrease each UAV's average moving distance, which saves energy. These benefits are brought by our proposed RGCN module, which learns UAV-user spatial correlation and navigates UAVs to pay attention to different students in need. Note that although GCRL-min(AoI) has



(a) $\bar{\kappa}$ (b) ψ (c) $\bar{\rho}$ (d) $\bar{\zeta}$

Fig. 4: Impact of no. of UAVs in Purdue.



(a) $\bar{\kappa}$ (b) ψ (c) $\bar{\rho}$ (d) $\bar{\zeta}$

Fig. 5: Impact of MCL in Purdue.

TABLE I: Impact of different hyperparameters.

		Purdue				NCSU				KAIST			
	N	$\bar{\kappa}$	ψ	$\bar{\rho}$	$\bar{\zeta}$	$\bar{\kappa}$	ψ	$\bar{\rho}$	$\bar{\zeta}$	$\bar{\kappa}$	ψ	$\bar{\rho}$	$\bar{\zeta}$
Square	2	8.964	0.892	0.358	0.809	20.894	0.772	0.105	0.821	10.579	0.790	0.280	0.852
	3	6.398	0.916	0.369	0.806	18.547	0.797	0.118	0.823	8.596	0.821	0.298	0.844
	4	6.403	0.909	0.369	0.807	21.132	0.753	0.101	0.826	9.215	0.814	0.286	0.842
	5	6.384	0.917	0.371	0.806	22.548	0.734	0.099	0.829	8.976	0.819	0.295	0.839
Gauss.	2	4.501	0.949	0.412	0.804	9.667	0.889	0.211	0.805	4.889	0.962	0.463	0.811
	3	**3.400**	0.977	0.433	0.802	**7.999**	0.918	0.227	0.802	**2.748**	0.986	0.472	0.803
	4	3.387	0.979	0.436	0.802	7.795	0.921	0.228	0.802	2.741	0.988	0.473	0.802
	5	3.384	0.981	0.437	0.801	7.690	0.923	0.228	0.802	2.739	0.988	0.475	0.801
Cosine	2	6.742	0.937	0.402	0.807	17.341	0.801	0.129	0.820	8.298	0.833	0.368	0.832
	3	5.124	0.941	0.409	0.807	15.365	0.823	0.149	0.817	6.235	0.895	0.389	0.820
	4	5.146	0.940	0.408	0.808	15.356	0.826	0.148	0.817	6.211	0.899	0.389	0.820
	5	5.175	0.940	0.408	0.809	16.976	0.824	0.148	0.817	6.209	0.901	0.393	0.819

already shown a good performance to meet the demand of students in three datasets, deploying only 2 UAVs is not enough, as confirmed in Fig. 4c, Fig. 6c and Fig. 8c.

C. Impact of Hyperparameters

We select three key hyperparameters from proposed RGCN module and MCTS structure, including (a) different similarity functions that determine the learning speed of UAV-user spatial correlation extraction, and (b) N as the maximum depth of MCTS. We fix $U = 2$. As shown in Table I, we see that all these hyperparameters yield a lowest point in terms of episodic AoI $\bar{\kappa}$. We observe that using cosine and square functions cannot bring equal AoI optimization effect as embedded gaussian function does. This is because the latter computes the relational response $\boldsymbol{\eta}$ on a UAV/user's features $\boldsymbol{z}$ as a weighted sum of features over all others in RGCN, which better expresses the UAV-user spatial correlation. Then, we see that the given a similarity function, increasing N from $N = 2$ will give lower episodic AoI $\bar{\kappa}$. This is because deeper tree search obviously brings more accurate value estimations. However, this improvement is quite limited when further improving N after $N = 3$. Furthermore, from Eqn. (17), the computational complexity will rise exponentially as N increases. On the other hand, Gaussian function produces better results than Square and Cosine similarity functions on average. Therefore, we choose Gaussian function with $N = 3$ as the best hyperparamters used in the following experiments.

TABLE II: Computational Complexity by time cost (ms).

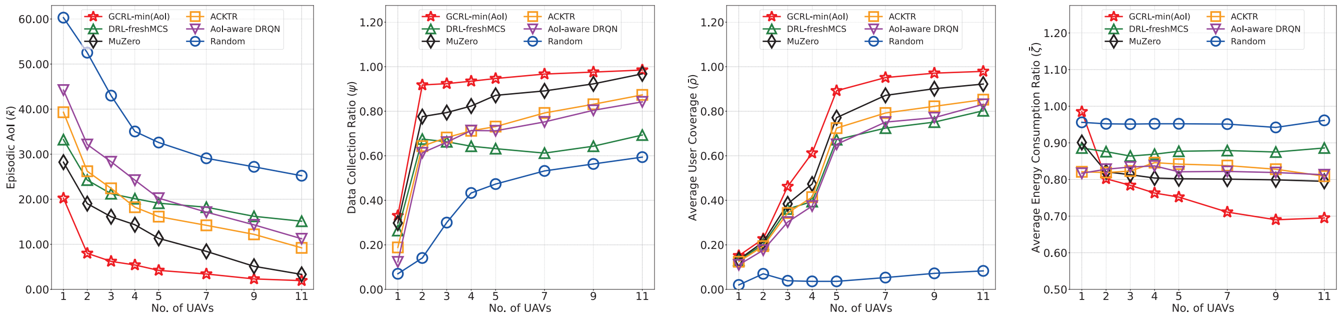
Method	Purdue	NCSU	KAIST
GCRL-min(AoI)	**7.998**	**5.677**	**11.003**
DRL-freshMCS	33.866	25.165	41.978
Muzero	9.867	6.993	15.497
ACKTR	15.223	11.035	23.778
AoI-aware DRQN	21.999	17.097	33.008

D. Computational Complexity Analysis

Computational complexity (by time cost) is shown in Table II. The running time to produce actions in a timeslot by CGRL-min(AoI) is much faster than other baselines, and slightly lower than another tree-based method Muzero. This is because we utilize RGCN instead of CNNs or RNNs for feature extractions, which is easy to utilize parallel CPU cores. That is, CGRL-min(AoI) do not use graphic card memory at all, but its time cost is still in the scale of millisecond, which is negligible and cost-effective in real UAV deployment where GPUs can be expensive and of big size.

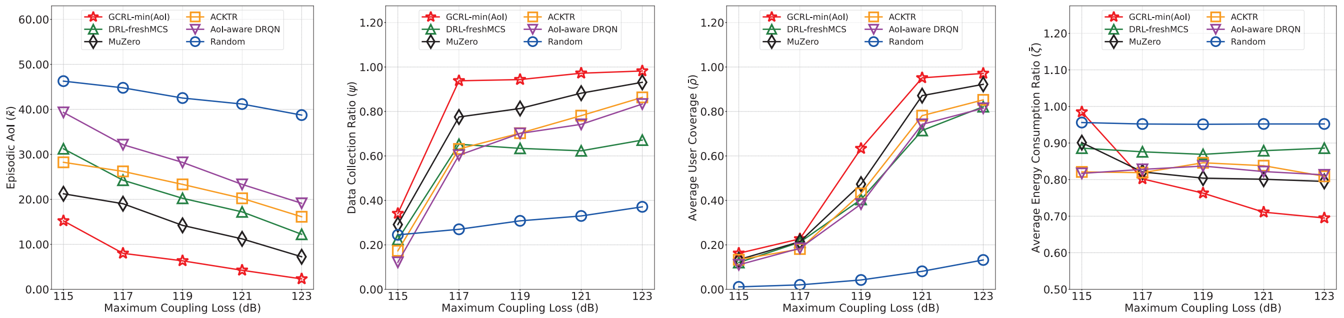
E. Ablation Study

The ablation study is performed by gradually removing two key components of our solution, i.e., RGCN and next state



(a) $\bar{\kappa}$ (b) ψ (c) $\bar{\rho}$ (d) $\bar{\zeta}$

Fig. 6: Impact of no. of UAVs in NCSU.



(a) $\bar{\kappa}$ (b) ψ (c) $\bar{\rho}$ (d) $\bar{\zeta}$

Fig. 7: Impact of MCL in NCSU.

prediction (pred). Results are shown in Table III. We see that the complete version CGRL-min(AoI) achieves 71%, 66% and 63% lower episodic AoI than GCRL-min(AoI) w/o RGCN in Purdue, NCSU and KAIST datasets, respectively. This confirms that RGCN successfully extracts the inter-correlations between UAVs and users more efficiently, especially when the environment (i.e., NCSU campus which have north/south parts connecting by two entrances that students go through) is more complicated (compared to Purdue and KAIST). Furthermore, GCRL-min(AoI) achieves 37%, 27% and 31% lower episodic AoI than GCRL-min(AoI) w/o pred in three datasets. This is because the model-based next state prediction can reduce the demand of experience data (than model-free methods) to train a good policy with accurate state value estimations.

TABLE III: Ablation study (when $U = 2$).

Dataset	Method	$\bar{\kappa}$	ψ	$\bar{\rho}$	$\bar{\zeta}$
Purdue	GCRL-min(AoI)	**3.400**	0.977	0.433	0.802
	GCRL-min(AoI) w/o RGCN	11.547	0.882	0.378	0.811
	GCRL-min(AoI) w/o pred	5.397	0.946	0.421	0.805
	GCRL-min(AoI) w/o RGCN, pred	12.837	0.878	0.365	0.809
NCSU	GCRL-min(AoI)	**7.999**	0.918	0.227	0.802
	GCRL-min(AoI) w/o RGCN	23.578	0.788	0.109	0.816
	GCRL-min(AoI) w/o pred	10.991	0.912	0.223	0.809
	GCRL-min(AoI) w/o RGCN, pred	25.001	0.772	0.103	0.820
KAIST	GCRL-min(AoI)	**2.748**	0.986	0.472	0.803
	GCRL-min(AoI) w/o RGCN	7.385	0.843	0.402	0.824
	GCRL-min(AoI) w/o pred	3.998	0.971	0.463	0.818
	GCRL-min(AoI) w/o RGCN, pred	9.458	0.799	0.375	0.831

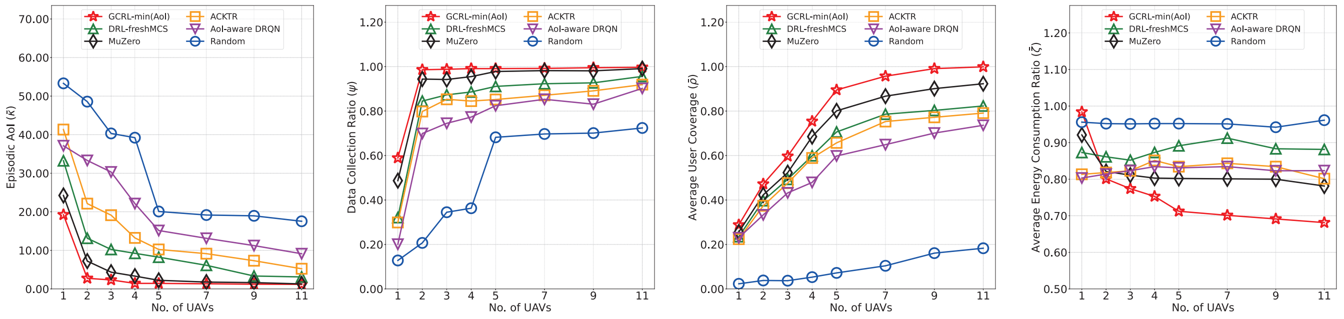
F. Comparing with Five Baselines

We compared GCRL-min(AoI) with five baselines in Purdue, NCSU and KAIST datasets on four previously introduced metrics when changing the number of UAVs and MCL.

- **DRL-freshMCS** [20]: It navigates a group of UAVs with multiple antennas to minimize AoI, which is considered as the state-of-the-art approach for AoI-aware MCS. In order to apply DRL-freshMCS in our UCS scenarios, we map our states into an 80×80 images, to fit the dimension of CNN inputs in DRL-freshMCS.
- **MuZero** [39]: It is the latest successor of MCTS (AlphaZero), which shows superior performance in Chess, Shogi and even Atari games. It learns to predict reward function and next state together, and updates DNN parameters every N steps periodically, which is considered as the state-of-the-art approach for tree-based DRL methods.
- **ACKTR** [40]: It is a well-known off-policy DRL approach in Atari benchmarks, using an approximate curvature called KFAC. For fair comparison, we add a classic GCN as a feature extraction module.
- **AoI-aware DRQN** [41]: It is another DRL-based solution for AoI-aware vehicle-to-vehicle networking, which considers high spatial mobility and temporally varying traffic information arrivals. Based on DRQN [42], it optimizes AoI in long-term tasks, which can directly run on our UCS scenarios by treating vehicles as mobile users.
- **Random**: We sample action $\boldsymbol{a}_t$ from $\mathcal{A}$ randomly.

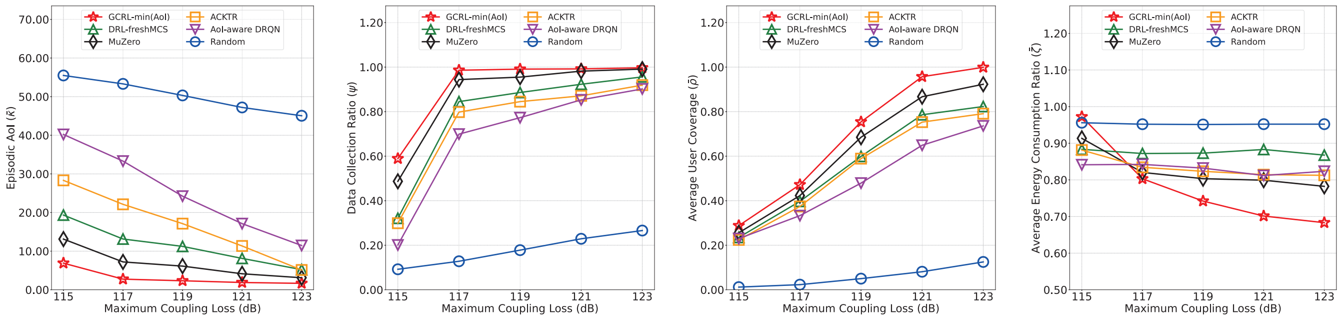
Results are shown in Fig. 4 – Fig. 9. We make three important observations.

GCRL-min(AoI) consistently outperforms all five baselines in terms of episodic AoI in three datasets. The reason is that DRL-freshMCS utilizes CNNs to extract spatial features



(a) $\bar{\kappa}$ (b) ψ (c) $\bar{\rho}$ (d) $\bar{\zeta}$

Fig. 8: Impact of no. of UAVs in KAIST.



(a) $\bar{\kappa}$ (b) ψ (c) $\bar{\rho}$ (d) $\bar{\zeta}$

Fig. 9: Impact of MCL in KAIST.

from a designed meta image but not consider the mobility and correlation between UAVs and users. ACKTR and AoI-aware DRQN both integrate UAVs' and users' features by simply mixing them as a high-dimensional vectors, which is insufficient for effective DRL training. MuZero performs better than the rest of baselines but still worse than our method. This is because MuZero predicts reward function and next state together, which needs more training episodes to coverage, since our reward function and UCS scenario is complex than common DRL benchmarks.

As shown in Fig. 4, Fig. 6 and Fig. 8, we set $MCL = 117$dB and show the impact of number of UAVs. With more employed UAVs, the attained episodic AoI keeps decreasing and even approaches to the lower bound 1 when $U = 11$. This is because more UAVs brings higher user coverage and data collection ratio, with lower energy consumption than fewer UAVs. In our UCS task with fixed task time, the total amount of data that need to be collected only depends on the number of users and their data collection capability, which is upper-bounded by $\sum_{m=1}^{M} T \cdot v^{m}_{\text{collect}} \cdot \tau$, as in Eqn. (5). Deploying more UAVS will help reduce the “collection burden” of each UAV, and thus promoting more efficient patterns of UAV cooperation. However, when $U > 5$, the gain of most methods becomes relatively minor in Purdue and KAIST datasets, because the data collection ratio saturates (to 1) in Fig. 4b and Fig. 8b. On the other hand, deploying more UAVs will directly expand the action space size, reducing the gap among different methods.

Results of varying MCL are shown in Fig. 5, Fig. 7 and Fig. 9, when we set $U = 2$. In all datasets, when increasing MCL, the episodic AoI keeps decreasing for most methods. This is because MCL influences the maximum coverage of certain radio access technology (we set MCL between 115dB and 123dB of 5G frequency 28GHz in this paper [22]). In our UCS task, the collected data upload and AoI update are both determined by MCL (see Eqn. (5) and (6)). Higher MCL allows each UAV to move a shorter distance but cover more users, which directly contributes to the optimization of episodic AoI.

VIII. Conclusion

In this paper, we proposed GCRL-min(AoI), a novel model-based DRL framework to navigate a group of UAVS as aerial BSs to meet the MCS demand of underlying mobile users, to minimize their attained AoI. Specifically, we proposed a novel model-based MCTS structure based on state-of-the-art approach “MCTS (AlphaZero)”. We improve it by adding a spatial UAV-user correlation extraction mechanism by a RGCN, and a next state prediction module to successfully reduce the use of experience data. Results on Purdue University, KAIST and NCSU campuses datasets confirm that GCRL-min(AoI) consistently outperforms five other baselines.

Acknowledgement

This paper was sponsored in part by National Natural Science Foundation of China (No. U21A20519 and 62022017), and in part by the National Research and Development Program of China under Grant 2019YQ1700. Corresponding author: C. H. Liu.

REFERENCES

[1] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G. Ahn, "The rise of people-centric sensing," *IEEE Internet Computing*, vol. 12, no. 4, pp. 12–21, 2008.

[2] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, "Uav trajectory planning for data collection from time-constrained iot devices," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 34–46, 2020.

[3] J. Li, H. Zhao, H. Wang, F. Gu, J. Wei, H. Yin, and B. Ren, "Joint optimization on trajectory, altitude, velocity, and link scheduling for minimum mission time in uav-aided data collection," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1464–1475, 2020.

[4] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *IEEE SECON'11*, 2011.

[5] Y. Hsu, E. Modiano, and L. Duan, "Age of information: Design and analysis of optimal scheduling algorithms," in *IEEE ISIT'17*, 2017, pp. 561–565.

[6] H. Chen, Q. Wang, Z. Dong, and N. Zhang, "Multiuser scheduling for minimizing age of information in uplink mimo systems," *IEEE ICCC'20*, 2020.

[7] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[8] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[10] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *AAAI'18*, 2018, pp. 3215–3222.

[11] C. H. Liu, Z. Chen, and Y. Zhan, "Energy-efficient distributed mobile crowd sensing: A deep learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1262–1276, 2019.

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

[13] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, and M. Guizani, "When mobile crowd sensing meets uav: Energy-efficient task assignment and route planning," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5526–5538, 2018.

[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 Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1274–1285, 2020.

[15] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021.

[16] Y.-P. Hsu, E. Modiano, and L. Duan, "Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals," *IEEE Transactions on Mobile Computing*, vol. 19, no. 12, pp. 2903–2915, 2020.

[17] H. Hu, K. Xiong, G. Qu, Q. Ni, P. Fan, and K. B. Letaief, "Aoi-minimal trajectory planning and data collection in uav-assisted wireless powered iot networks," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1211–1223, 2021.

[18] M. Chen, W. Liang, and S. K. Das, "Data collection utility maximization in wireless sensor networks via efficient determination of uav hovering locations," in *IEEE PerCom'21*, 2021, pp. 1–10.

[19] Y. Li, W. Liang, W. Xu, Z. Xu, X. Jia, Y. Xu, and H. Kan, "Data collection maximization in iot-sensor networks via an energy-constrained uav," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.

[20] 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 *IEEE INFOCOM'21*, 2021, pp. 1–10.

[21] J. G. Andrews, T. Bai, M. N. Kulkarni, A. Alkhateeb, A. K. Gupta, and R. W. Heath, "Modeling and analyzing millimeter wave cellular systems," *IEEE Transactions on Communications*, vol. 65, no. 1, pp. 403–430, 2017.

[22] M. R. Akdeniz, Y. Liu, M. K. Samimi, S. Sun, S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter wave channel modeling and cellular capacity evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1164–1179, 2014.

[23] M. Gapeyenko, I. Bor-Yaliniz, S. Andreev, H. Yanikomeroglu, and Y. Koucheryavy, "Effects of blockage in deploying mmwave drone base stations for 5g networks and beyond," in *IEEE ICC'18 Workshops*, 2018, pp. 1–6.

[24] 3GPP, "5G; study on scenarios and requirements for next generation access technologies," TR 36.913, 5 2017, version 14.2.0.

[25] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[26] J. Oh, S. Singh, and H. Lee, "Value prediction network," in *NeurIPS'17*, 2017.

[27] H. Zhao, Q. She, C. Zhu, Y. Yang, and K. Xu, "Online 3d bin packing with constrained deep reinforcement learning," *AAAI'20*, 2020.

[28] R. S. Sutton, "Dyna, an integrated architecture for learning, planning and reacting," in *Working Notes of the AAAI Spring Symposium*, 1991.

[29] G. Z. Holland, E. J. Talvitie, and M. Bowling, "The effect of planning shape on dyna-style planning in high-dimensional state spaces," *arXiv preprint arXiv:1806.01825*, 2018.

[30] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, "Relational graph learning for crowd navigation," in *IROS'20*, 2020.

[31] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR'17*, 2017.

[32] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *IEEE CVPR'18*, 2018, pp. 7794–7803.

[33] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *PMLR'10*, 2010, pp. 249–256.

[34] P.-A. Coquelin and R. Munos, "Bandit algorithms for tree search," *arXiv preprint cs/0703062*, 2007.

[35] B. Ottens, C. Dimitrakakis, and B. Faltings, "Duct: An upper confidence bound approach to distributed constraint optimization problems," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 5, pp. 1–27, 2017.

[36] H. Zhang, M. A. Roth, R. K. Panta, H. Wang, and S. Bagchi, "Crowdbind: Fairness enhanced late binding task scheduling in mobile crowdsensing," *the 17th International Conference on Embedded Wireless Systems and Networks (EWSN)*, pp. 1–12, 2020.

[37] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong, "CRAWDAD dataset ncsu/mobilitymodels (v. 2009-07-23)," Downloaded from https://crawdad.org/ncsu/mobilitymodels/20090723, Jul. 2009.

[38] DJI, "Matrice 600 pro - product information - dji," https://www.dji.com/cn/matrice600-pro/info#specs.

[39] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, and T. Graepel, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, p. 604–609, 2020.

[40] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba, "Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation," *NeurIPS' 17*, vol. 30, pp. 5279–5288, 2017.

[41] X. Chen, C. Wu, T. Chen, H. Zhang, Z. Liu, Y. Zhang, and M. Bennis, "Age of information aware radio resource management in vehicular networks: A proactive deep reinforcement learning perspective," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2268–2281, 2020.

[42] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," in *2015 AAAI fall symposium series*, 2015.