QoI-Aware Mobile Crowdsensing for Metaverse by Multi-Agent Deep Reinforcement Learning

Yuxiao Ye, Hao Wang, Chi Harold Liu, *Senior Member, IEEE*, Zipeng Dai, Guozheng Li, Guoren Wang, and Jian Tang, *Fellow, IEEE*

***Abstract*—Metaverse is expected to provide mobile users with emerging applications both in regular situation like intelligent transportation services and in emergencies like wireless search and disaster response. These applications are usually associated with stringent quality-of-information (QoI) requirements like throughput and age-of-information (AoI), which can be further guaranteed by using unmanned aerial vehicles (UAVs) as aerial base stations (BSs) to compensate the existing 5G infrastructures. In this paper, we consider a new QoI-aware mobile crowdsensing (MCS) campaign by UAVs which move around and collect data from mobile users wearing metaverse devices. Specifically, we propose "MetaCS", a multi-agent deep reinforcement learning (MADRL) framework with improvements on a Transformer-based user mobility prediction module between regions and a relational graph learning mechanism to enable the selection of most informative partners to communicate for each UAV. Extensive results and trajectory visualizations on three real mobility datasets in NCSU, KAIST and Beijing show that MetaCS consistently outperforms six baselines in terms of overall QoI index, when varying different numbers of UAVs, throughput requirement, and AoI threshold.**

***Index Terms*—Mobile crowdsensing for metaverse, Quality-of-information, User behavior modeling, Multi-agent deep reinforcement learning**

I. Introduction

Metaverse is expected to revolutionize the way people interact by building immersive environment to communicate via digital avatars from anywhere at anytime [1]–[3]. In metaverse, human players wearing virtual reality (VR) or augmented reality (AR) devices like headsets and helmets (e.g., Oculus and HoloLens [4]) to interact with virtual immersive 3D environments to provide cyber-virtual experiences in physical worlds. They constantly travel on-the-go to achieve certain tasks (e.g., virtual healthcare and transportation services) via 5G ultra-high reliable and ultra-low latency wireless connections. User behaviors (e.g., trajectories and postures) are uploaded to the surrounding mobile edge computing (MEC) infrastructures and base stations (BSs) in real-time, and the object creation, scene rendering and others for immersive 3D interactions in virtual environments are periodically delivered to their equipped devices.

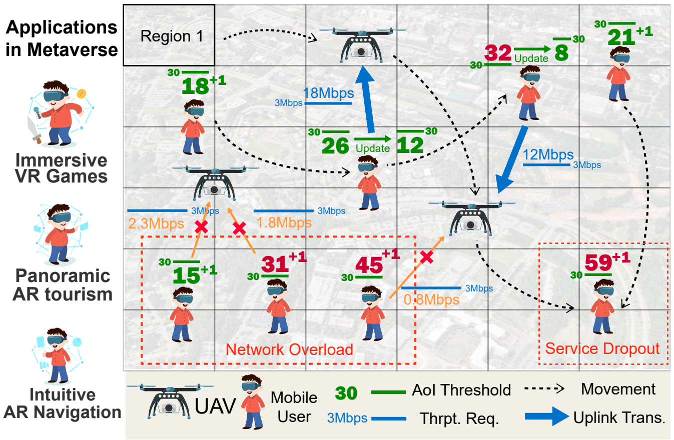


Fig. 1: Overall considered scenario of QoI-aware MCS for metaverse.

However, existing terrestrial MEC infrastructures in a 5G network may be overloaded so that the network resources (e.g., bandwidth and time) are insufficient to service all users simultaneously. Furthermore, MEC infrastructures and BSs usually locate at fixed positions so that they cannot meet the application-level quality-of-information (QoI) requirement of underlying serviced users dynamically, whose spatial density are continuously changing. QoI represents a set of metrics to judge if information is fit-for-use for particular purpose [5]. Here we define the QoI requirements of a metaverse application by (but are not limited to) the network throughput (in Mbps) and data freshness from the time it is generated till received. Data freshness requirement for QoI can be represented by the "age of information" (AoI [6], [7]) 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, within a given AoI threshold of metaverse applications.

Mobile crowdsensing (MCS [8], [9]) paved the way for possible solutions to provide wide-range, timely metaverse applications in the above resource limited scenarios. The authors in [10], [11] envisioned a MCS-enabled metaverse paradigm for digital twins and immersive VR/AR applications, where unmanned aerial vehicles (UAVs) are employed in temporary adventures as mobile BSs [12]–[14], due to their features of high flexibility and ease of deployment [15], [16], to provide sufficient amount of data to satisfy metaverse applications with frequent transient data demands (e.g., seasonal festival activities requested by virtual sightseeing providers). However, specific solutions are not given. Thus, we explicitly consider the QoI-aware MCS campaign for metaverse in this paper,

Y. Ye, H. Wang, C. H. Liu, Z. Dai, G. Li and G. Wang are with School of Computer Science and Technology, Beijing Institute of Technology, China. Email: chiliu@bit.edu.cn. J. Tang is with Midea Group. Y. Ye and H. Wang contributed equally to this work. Corresponding author: C. H. Liu.

This paper was sponsored by National Natural Science Foundation of China (No. U21A20519 and 62022017).

where multiple UAVs are deployed and navigated as mobile aerial BSs to provide uplink network capacities of underlying metaverse applications across space and time domains. As shown in Fig. 1, UAVs move around in the workzone and receive data uploaded from multiple users (e.g., students and staff members in campus environments) who are equipped with metaverse devices.

The key challenge is how to balance the trade-off between UAV movements for data collection, saving battery power and maintaining satisfactory level of QoI for all users simultaneously. For example, students in a campus environment usually exhibit group behaviors (e.g., taking classes and going to the dining hall together), where the terrestrial MEC network will be overloaded and thus more UAVs will be needed. On the other hand, if a user goes to a corner area without being serviced, AoI threshold will be violated and thus service dropout may occur. In this case, a UAV should be scheduled to access this user in time even if long-distance travel is required.

Therefore, the goal of this paper is to navigate a group of UAVs as mobile BSs to ensure the satisfactory attained level of QoI for underlying metaverse applications with limited energy supply. To calculate the optimal policy, it is hard to formulate it as a constrained optimization problem and utilize classical mathematical methods (e.g., Lagrangian duality with KKT conditions) to obtain a closed-form solution. Recent achievements along the direction of multi-agent deep reinforcement learning (MADRL [17], [18]) offer a promising way to address this problem. However, existing approaches seldom consider suitable collaborative control mechanism among UAVs and accurate prediction of the spatial-temporal user distribution.

To this end, in this paper, we propose "MetaCS", a QoI-aware MCS framework for metaverse, by using the state-of-the-art MADRL solution IPPO [19] as the start point of the design. Our contribution is four-fold:

1) To meet QoI requirements of metaverse applications by MCS, we explicitly define two metrics as throughput satisfactory ratio and AoI satisfactory ratio, and integrate them together as one single performance metric called "overall QoI index", to indicate to what extent the provided metaverse service can meet the application requirement.
2) We propose a human mobility prediction mechanism by designing a novel Transformer-based framework to model the spatiotemporal moving patterns of mobile users between partitioned regions in the workzone, which is especially helpful for UAVs to forecast and deal with surrounding emergencies that affect overall QoI experience by incurring network overload or service dropout.
3) We propose a collaborative UAV control framework by designing a relational graph learning mechanism that enables the selection of most informative partners for each UAV, and a graph-guided scheme to achieve efficient coordination of UAVs based on IPPO.
4) We perform extensive experiments on three real-world mobility datasets in NCSU (USA) and KAIST (South Korea) for campus environments, and Beijing (China) for urban environments. We find the most appropriate hyperparameters, visualize the trajectories and make performance comparisons with six baselines. Results confirm that MetaCS has robust improvements on overall QoI index.

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

II. Related Work

A. MCS and UAV Crowdsensing

MCS leverages mobile crowds to provide large-scale, low-latency, and high-quality sensing services in metaverse [20]. For example, Zhou *et al.* in [21] proposed a two-step solution for joint controlling of sensing and transmission processes and corresponding energy consumption in MCS, which are validated in high-capacity IoT sensing scenarios. Li *et al.* in [22] developed a Context-Aware Worker Selection (CAWS) algorithm to select a large number of workers to perform a sensing task collaboratively under a limited budget.

On the other hand, UAVs, forming as flying ad-hoc networks, are able to provide cost-effective MCS services, to offer high flexibility and easy adaptability to different application contexts, making them suitable to meet crucial sensing requirements [23]–[25]. For example, Gao *et al.* in [21] optimized sensing coverage and data quality by guiding human participants and using UAVs to collect data from rarely sensed points of interest. Liu *et al.* in [12] proposed a MADRL framework for energy-efficient multi-UAV navigation to maximize the total amount of collected data and ensure geographical fairness among randomly distributed points of interest. Hu *et al.* in [26] introduced an optimization problem for the wireless-powered IoT system, and utilized UAVs to transfer energy and collect data from ground sensor nodes.

B. MADRL and Using IPPO as Base Design

In cooperative tasks, a group of agents interacts with the environment to learn to achieve the shared objective in a collaborative way. In general, the task is modeled as a decentralized partially observable Markov decision process (Dec-POMDP) [27], as a tuple $< \mathcal{U}, \mathcal{S}, \mathcal{O}, \mathcal{A}, R, Pr, \gamma >$, where $\mathcal{U}, \mathcal{S}, \mathcal{O}, \mathcal{A}$ are the set of agents, states, observations and actions, respectively. R, Pr, γ are the reward function, state transition probabilities, discount factor, respectively. The shared objective is to maximize the expected discounted cumulative reward $\mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t \sum_{u=1}^{U} r_t^u\right]$.

To solve it, various MADRL methods are proposed. For example, IPPO [19] and DNAC [28] completely separated the observations of agents and independently updated the corresponding critics. Moreover, HiT-MAC [29] and EOI [30] encouraged agent cooperation by setting progressive goals for each agent or providing auxiliary rewards. However, they cannot fully exploit each agent observation and thus are vulnerable to the non-stationary cases caused by policy changes across

TABLE I: Important notations used in this paper.

Notation	Explanation
$u, U, \mathcal{U}, P$	Index, total number, set of UAVs and number of antennas on each UAV
$m, M, \mathcal{M}, \mathcal{M}_t^u$	Index, total number, set of mobile users and set of scheduled users
$t, T, \delta, \delta_{t,\text{mov}}^u, \delta_{t,\text{col}}^u$	Index, total number of timeslots, duration of a timeslot, duration for UAV movement and data collection in each timeslot
$d_t^m, \Delta d_t^{u,m}, D$	Current data amount of user m, and collected data and generated data amount in a timeslot
$\mu_t^{u,m}, \mu_{\text{th}}^m, \tau_{\text{th}}^m$	Data rate of a user in a timeslot, throughput requirement, AoI threshold of a user m
τ, I_μ, I_τ, I	Episodic AoI, throughput satisfactory ratio, AoI satisfactory ratio, overall QoI index
$\boldsymbol{o}_t^u, \boldsymbol{a}_t^u, r_t^u$	Observation, action and reward for UAV u in timeslot t

agents. In our considered scenario where network overload or service dropout may happen, observation exchanges and collaborative execution among UAVs become even more critical. Another line of methods aggregated all the observations and guided agents to collaborate through a shared reward with credit assignment, e.g., MAPPO [31] and DPN [32]. In our scenario, when more UAVs are deployed, these methods may suffer from insufficient network bandwidth by increased communication overhead, and individual UAV contribution to the shared reward cannot be clearly distinguished resulting in a sub-optimal policy. To this end, we choose IPPO as the start point of our design, however it is worth noting that our proposed framework MetaCS can also be applied to other actor-critic based MADRL solutions.

III. System Model

Without loss of generality, we assume that a group of UAVs $\mathcal{U} \triangleq \{u|u = 1, 2..., U\}$ are flying in a 2D cartesian coordinate system at a fixed height to collect data from mobile users equipped metaverse devices (e.g. Google Glass[1]), denoted by $\mathcal{M} \triangleq \{m|m = 1, 2, ..., M\}$. These devices continuously generate user data (e.g., position calibration in the virtual world). Each UAV is equipped with P antennas, and each device is with one single omni-directional antenna.

We discretize a task into T timeslots with a fixed length δ, where t represents the index of the current timeslot. In each timeslot, δ is divided into two parts: for UAV movement $\delta_{t,\text{mov}}^u$ and for user data upload in the uplink channel $\delta_{t,\text{col}}^u$. For the former, UAV u moves in a radial direction $\theta_t^u \in [0, 2\pi)$ with a fixed moving speed $v_t^u \in [0, v_{\max}]$. For the latter, we have $\delta_{t,\text{col}}^u = \delta - \delta_{t,\text{mov}}^u$.

All UAVs has equal initial energy reserve e_0. Based on the propulsion energy model of rotary-wing UAVs [33], the energy consumption of a UAV u in t is given by:

$$\mathrm{e}_t^u = \tau \cdot \left[c_1\left(1 + \tfrac{3\cdot(v_t^u)^2}{(v_{\text{tip}})^2}\right) + c_2\left(\sqrt{1 + \tfrac{(v_t^u)^4}{4\bar{v}^4}} - \tfrac{(v_t^u)^2}{2\bar{v}^2}\right)^{\frac{1}{2}} + \tfrac{1}{2}c_3(v_t^u)^3\right], \tag{1}$$

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. Thus the average energy consumption of all UAVs after task completion can be calculated by $\bar{e} = \frac{1}{U}\sum_{u=1}^{U}\sum_{t=1}^{T} e_t^u$.

Similar to [34], we consider that each UAV can collect data from multiple users (no more than P) simultaneously by adopting Multi-User Multiple-Input Multiple-Output (MU-MIMO) techniques. In each timeslot t, each user m generates a fixed-size data D, and his/her current data amount in the buffer is denoted by $d_t^m = d_{t-1}^m + D$. It is crucial to carefully decide which group of users will be scheduled for uplink transmission in t that has high impact on the overall performance. Thus, for each UAV u, let $\mathcal{M}_t^u \triangleq \{m \in \mathcal{M}|$ a user m is selected by the UAV $u\}$ denote the set of transmitting users and $M_t^u = |\mathcal{M}_t^u|$. Then, the received signal vector $\boldsymbol{\varrho}_t^u$ of size M_t^u can be expressed as:

$$\boldsymbol{\varrho}_t^u = \sqrt{\rho_0}\mathbf{H}_t^u\boldsymbol{\eta}_t^u + \boldsymbol{c}_t, \tag{2}$$

where $\mathbf{H}_t^u$ represents corresponding channel matrix between a UAV u and selected M_t^u users of size $P \times M_t^u$. $\sqrt{\rho_0}\boldsymbol{\eta}_t^u$ is the vector of symbols of size $M_t^u \times 1$ which are transmitted simultaneously by these users (where the average transmitted power of each user is ρ_0). $\boldsymbol{c}_t \sim \mathcal{CN}(\mathbf{0}, \psi_0\mathbf{I}_P)$ represents the additive white Gaussian noise at each UAV, with noise power ψ_0.

Without loss of generality, we model the ground-to-air uplink propagation channel by considering both large-scale and small-scale channel fading. Specifically, we jointly consider the Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) effects, along with their occurrence probabilities. Following [35], we have path loss:

$$\begin{aligned}\psi_t^{u,m} &= 20\log\left(\frac{4\pi f l_t^{u,m}}{c}\right) + w_{\text{NLoS}} \\ &+ \frac{w_{\text{LoS}} - w_{\text{NLoS}}}{1 + w_1\exp\left(-w_2(\theta_t^{u,m} - w_1)\right)},\end{aligned} \tag{3}$$

where $l_t^{u,m}, \theta_t^{u,m}$ denote the Euclidean distance and the elevation angle between a UAV u and a user m, f is the channel frequency, c is the speed of light, w_{LoS} and w_{NLoS} are average additional path loss, w_1 and w_2 are constants that depend on the environment (i.e., rural or urban), respectively. Then, large-scale fading is calculated by $\beta_t^{u,m} = \beta_0\psi_t^{u,m}$, where β_0 is the channel gain at a reference distance of one meter.

For small-scale fading, we consider the F-factor Rician fading with $\mathbb{E}[\|\boldsymbol{\vartheta}_t^{u,m}\|^2] = 1$ for both LoS and NLoS links during uplink transmissions, as: $\boldsymbol{\vartheta}_t^{u,m} = \sqrt{\frac{F}{F+1}}\bar{\boldsymbol{\vartheta}}_t^{u,m} + \sqrt{\frac{1}{F+1}}\tilde{\boldsymbol{\vartheta}}_t^p$, where $\bar{\boldsymbol{\vartheta}}_t^{u,m}$ is the complex vector of the LoS path between a user m and P antennas of a UAV u, while $\tilde{\boldsymbol{\vartheta}}_t^p \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_P)$ denotes the Rayleigh fading component for NLoS links.

We consider linear beamforming [36] from users to a UAV, where the received data is:

$$\hat{\boldsymbol{\eta}}_t^u = [\mathbf{W}_t^u]^{\mathrm{H}}\boldsymbol{\varrho}_t^u = \sqrt{\rho_0}[\mathbf{W}_t^u]^{\mathrm{H}}\mathbf{H}_t^u\boldsymbol{\eta}_t^u + [\mathbf{W}_t^u]^H\boldsymbol{c}_t, \tag{4}$$

where $\mathbf{W}_t^u = \left[\boldsymbol{w}_t^{u,\mathcal{M}_t^u(1)}, \cdots, \boldsymbol{w}_t^{u,\mathcal{M}_t^u(M_t^u)}\right]$ denotes the beamforming matrix with the same size as $\mathbf{H}_t^u$, and $\|\boldsymbol{w}_t^{u,m}\| = 1, m \in \mathcal{M}_t^u$, respectively. To eliminate the interference among different users, we employ Zero-Forcing [37], i.e, $\mathbf{W}_t^u =$

[1] https://veo.glass/en_US/products/google-glass

$\mathbf{H}_t^u[(\mathbf{H}_t^u)^{\mathrm{H}}\mathbf{H}_t^u]^{-1}$. Then, we have the received signal-to-noise ratio (SNR) from a user m to a UAV u in timeslot t, as: $\gamma_t^{u,m} = \frac{\rho_0 \|(\boldsymbol{w}_t^{u,m})^{\mathrm{H}}\mathbf{H}_t^u(m)\|^2}{\psi_0}$. To measure the actual data rate $\mu_t^{u,m}$, we apply the Wishart matrix [36] and Jensen's inequality [34], to approximate the data rate as:

$$\begin{aligned}\mu_t^{u,m} &= B\log(1+\gamma_t^{u,m}) \\ &= B\log\left(1+\frac{\rho_0}{\mathbb{E}\left\{[((\mathbf{H}_t^u)^{\mathrm{H}}\mathbf{H}_t^u)^{-1}]_{p,p}\right\}\psi_0}\right) \\ &\geq B\log\left(1+\frac{\rho_0\beta_t^{u,m}}{\frac{\psi_0}{P-M_t^u}}\right) \\ &= B\log\left(1+\frac{(P-M_t^u)\rho_0\beta_0\psi_t^{u,m}}{\psi_0}\right),\end{aligned} \tag{5}$$

where B is the available bandwidth. Note that here we use Maximum-Ratio Combining (MRC [38]) beamforming (i,e., $\mathbf{W}_t^u = \mathbf{H}_t^u$) instead of ZF, in special cases where $M_t^u = 1$, thus:

$$\mu_t^{u,m} \triangleq B\log\left(1+\frac{\Omega_t^u\rho_0\beta_0\psi_t^{u,m}}{\psi_0}\right), \tag{6}$$

where $\Omega_t^u = P - M_t^u$ if $M_t^u \geq 2$, and $\Omega_t^u = P$ if $M_t^u = 1$. Following [39], we make the assumption that successful transmissions of D can only occur if the data rate exceeded the throughput requirement μ_{th}^m and at least one piece of data is transmitted within the collection time of each UAV u, as: $\mu_t^{u,m} \geq \mu_{\text{th}}^m$ and $\mu_t^{u,m} \cdot \delta_{t,\text{col}}^u \geq D$.

Finally, each user also maintains a First-In-First-Out (FIFO) queue and the data leave the queue only when UAVs approach and collect it. In each timeslot t, a UAV u spends $\delta_{t,\text{col}}^u$ of time staying as a temporal multi-antenna BS to collect data from at most M_t^u users simultaneously by uplink MU-MIMO, with different data rates $\mu_t^{u,m}$. After, the remaining data at user m becomes $d_t^m - \sum_u \Delta d_t^{u,m}$, where $\Delta d_t^{u,m} = \min\left(d_t^m, \lfloor\frac{\mu_t^{u,m}\delta_{t,\text{col}}^u}{D}\rfloor\right)$ represents data collected by UAV u in timeslot t. Note that when a user is servived by multiple UAVs simultaneously, he/she will transmit the same data to all the UAVs via their equipped omni-directional antenna, which enhances the robustness of data collection process.

IV. Problem Definition and Formulation

The goal of this paper is to ensure the attained QoI (in terms of throughput and data freshness requirements) for all users in metaverse. For the former, we aim to provide satisfactory amount of bandwidth to meet his/her throughput requirement to enable user uplink data transmissions. For the latter, we use AoI [40] to capture the data freshness and explicitly define an AoI threshold τ_{th}^m to represent the maximum waiting time that users can tolerate.

Definition 1: Throughput Satisfactory Ratio. It is defined as the ratio between the attained throughput μ_T^m during the task process T and the throughput requirement μ_{th}^m enforced by the metaverse application, of all users, as:

$$I_\mu = \frac{1}{M}\sum_{m=1}^{M} I_\mu^m = \frac{1}{M}\sum_{m=1}^{M}\frac{\mu_T^m}{\mu_{\text{th}}^m} \in [0,1], \tag{7}$$

where $\mu_T^m = \frac{1}{T\delta}\sum_{u=1}^{U}\sum_{t=1}^{T}\Delta d_t^{u,m}$ denotes the average data rate when transmitting the total amount of data to UAVs during the task duration, and μ_{th}^m is determined by the specific metaverse application of a user m. I_μ represents the average degree of satisfaction the MCS-enabled metaverse system can provide to the end users in terms of throughput, with a higher value of I_μ closer to 1 being preferable, and values higher than 1 is clipped since it is fully satisfied and providing more goes beyond the application requirement.

Definition 2: Episodic AoI. It is defined as the average waiting time of the earliest generated data not yet uploaded at all users' FIFO queues at each timeslot t, by:

$$\tau = \frac{1}{MT}\sum_{m=1}^{M}\sum_{t=1}^{T}\tau_t^m = \frac{1}{MT}\sum_{m=1}^{M}\sum_{t=1}^{T}(t - t_1^m), \tag{8}$$

where t_1^m denotes the timeslot index of the earliest generated data in a user m's FIFO queue (that has not been uploaded yet). Thus, a smaller τ represents a shorter average waiting time or better metaverse application experience for user uplink data transmissions.

Since different metaverse applications requires diverse AoI performance, we explicitly introduce an AoI threshold τ_{th}^m to represent the maximum tolerable waiting time, or an acceptable bar for user uplink data transmissions. To capture the essence that UAVs need to maintain the satisfactory level of attained AoI of all users at every timeslot, we introduce the following definitions.

Definition 3: AoI Satisfactory Ratio. It is defined as the average proportion of time during which a user's AoI threshold is successfully maintained, over the course of their task duration T, of all users, as:

$$I_\tau = \frac{1}{MT}\sum_{m=1}^{M}\sum_{t=1}^{T} I_{t,\tau}^m = \frac{1}{MT}\sum_{m=1}^{M}\sum_{t=1}^{T}\mathbf{1}(\tau_t^m \leq \tau_{\text{th}}^m) \in [0,1], \tag{9}$$

whose value closer to 1 indicates the higher possibility to provide end user satisfactory AoI experiences; $\mathbf{1}(\cdot)$ is the indicator function, where it takes value 1 if the condition is satisfied and otherwise 0.

Since our goal is to ensure both throughput and data freshness as the overall QoI requirement, we aim to define an integrated performance index to consider these two simultaneously, as the minimum of two ratios. In this way, a user's experience is quantified. Further, energy consumption of all UAVs need to be fully considered, and thus we introduce the overall QoI index as:

$$I = \frac{\min(I_\mu, I_\tau)}{\bar{e}}. \tag{10}$$

A. Problem Definition

Our goal is to find an optimal to navigate all UAVs and to schedule all users' uplink transmissions to maximize the overall QoI index in T timeslots, as the following optimization problem:

$$\text{P1}: \max_{\{v_t^u,\theta_t^u,\mathcal{M}_t^u\}} I \quad \textbf{s.t.} \sum_{t=1}^{T} e_t^u \leq e_0, \quad \forall u \in \mathcal{U}. \tag{11}$$

Note that P1 is challenging to solve due to the following reasons. First, two time periods $\delta^u_{t,\text{mov}}, \delta^u_{t,\text{col}}$ in a timeslot t are trading off with each other thus obtaining optimal time allocation is challenging. Second, QoI requirements for throughput and data freshness vary from different metaverse applications. For example, tasks with extensive VR and AR requests may pose greater challenges in mitigating network overload and AoI violation, respectively [41]. Thus it is challenging to satisfy all the users with diverse QoI requirements at the same time.

Third, satisfactory episodic AoI experiences in a task duration is trading-off with the received throughput of all users, under the limited UAV capabilities and total network bandwidth. On one hand, when UAVs are evenly scattered across different regions to achieve spatial division of labor, episodic AoI attained by all users may violate the AoI threshold given that insufficient number of UAVs are scheduled to provide satisfactory uplink throughput for usually unevenly distributed users. Contrarily, when UAVs well learn to collaborate to service users simultaneously by OFDMA in the same region, sufficient throughput can be guaranteed, however possibly sacrificing AoI experiences of those remote users. Thus, an optimal UAV movement is expected to ensure two QoI metrics at the same time.

Finally, selecting an optimal set of users to service by a UAV in a timeslot is challenging. According to Eqn. (6), the uplink MU-MIMO data rate $\mu^{u,m}_t$ relates to the number of scheduled users M^u_t and channel conditions from a user m to a UAV u. An AoI greedy policy may schedule as many users as possible that likely increase the AoI satisfactory ratio, however bringing lower $\mu^{u,m}_t$ and unsuccessful data upload, and thus lower throughput satisfactory ratio as a return.

The above optimization problem is NP-hard since even checking the optimum requires a thorough search of the entire solution space, where computational complexity will exponentially increase with respect to task duration T and UAVs U. Hence, a good heuristic trajectory planning strategy is required. We model P1 as a sequential decision problem and utilizing MADRL methods to solve it.

B. Problem Formulation as a Dec-POMDP

Let a seven-tuple $< \mathcal{U}, \mathcal{S}, \mathcal{O}, \mathcal{A}, R, Pr, \gamma >$ be a Dec-POMDP for P1, where Pr and γ are transition probabilities and discount factor, respectively.

1) Observation space: $\mathcal{O} \triangleq \{\boldsymbol{o}_t\}$. Each UAV maintains its local observation $\boldsymbol{o}^u_t$ with a fixed sensing range, which consists of two parts. First is the location, remaining amount of data, and data generation time of all users within the sensing range; and second is the current position and remaining energy of all UAVs within the sensing range.

2) Action space: $\mathcal{A} \triangleq \{\boldsymbol{a}_t\}$. For each UAV, its $\boldsymbol{a}^u_t = (v^u_t, \theta^u_t, \mathcal{M}^u_t)$, where v^u_t is bounded by a maximum speed $v_{\max}$; θ^u_t represents the angle which controls the direction of UAV movement; and $\mathcal{M}^u_t$ is its scheduled users to service in a timeslot.

3) Reward function: For each UAV u, its environmental reward is:

$$r^u_t = \frac{\min(I^u_{t,\mu}, I^u_{t,\tau})}{e^u_t}, \quad \forall t, u, \tag{12}$$

where $I^u_{t,\mu}$ and $I^u_{t,\tau}$ denote the attained temporal throughput and AoI satisfactory ratios in timeslot t, similar to the definition of Eqn. (9) but defined from a UAV u's perspective, as:

$$I^u_{t,\mu} = \frac{\sum_{m\in\mathcal{M}^u_t} \mu^m_t}{\sum_{m\in\mathcal{M}^u_t} \mu^m_{\text{th}}},\ I^u_{t,\tau} = \frac{\sum_{m\in\mathcal{M}^u_t} \mathbf{1}(\tau^m_t \leq \tau^m_{\text{th}})}{M^u_t},\ \forall t, u. \tag{13}$$

It is worth noting that maximizing Eqn. (12) in this Dec-POMDP settings is strictly equivalent to minimizing Eqn. (11), which is compatible with MADRL-based methods as the start point of our design.

V. Proposed Solution: MetaCS

We propose "MetaCS", a novel MADRL framework to navigate a group of UAVs to provide satisfactory QoI services for mobile users in metaverse. As shown in Fig. 2, MetaCS consists of two improvements: a human-centric predictive framework called "**Pred**" to forecast mobile users behaviors in the task workzone (see Section V-A), and a collaborative multi-UAV control framework called "**Colla**" by relational graph-guided IPPO [19] (see Section V-B).

A. Human-Centric Predictive Framework for Mobile User Behavior Prediction

We propose a human-centric behavior prediction framework **Pred** by modeling the spatial-temporal movement patterns of mobile users. First, we divide the entire workzone into Z regions, and without loss of generality, we consider a common phenomenon that users in the same region typically exhibit similar behavior patterns in terms of the in/out flows z to/from nearby regions. We define z as the number of users moving into/out of the corresponding region during timeslot t, denoted by $x^z_{t,\text{in}}$ and $x^z_{t,\text{out}}$, respectively. Thus, $\boldsymbol{x}_t = \{x^z_{t,\text{in}}, x^z_{t,\text{out}}\}^Z_{z=1} \in \mathbb{R}^{Z\times 2}$ represents inflows and outflows of all the regions during timeslot t.

Then, given the sequence of in/outflows of all the regions in the past L timeslots $\mathbf{X} = \{\boldsymbol{x}_{t-L+1}, \cdots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t\}$, **Pred** aims to accurately predict $\boldsymbol{x}_{t+1}$ in the next timeslot. For an input matrix $\mathbf{X} \in \mathbb{R}^{L\times Z\times 2}$, we use $\mathbf{X}_{t::} \in \mathbb{R}^{Z\times 2}$ to represent the temporal slice (i.e., different timeslots) and $\mathbf{X}_{:z:} \in \mathbb{R}^{L\times 2}$ to represent the spatial slice (i.e., different regions), respectively. Leveraging Transfromer [42] framework that is widely used to model complex and dynamic spatial-temporal dependencies effectively, our proposed **Pred** is composed of the skip connection of multiple spatial-temporal encoder layers. Each encoder contains both a spatial and a temporal component, namely: a Temporal Self-Attention (TSA) module for long-term temporal pattern modeling, and a Multi-Head Spatial Self-Attention (MHSSA) module that simultaneously captures the short-range and long-range dynamic spatial dependencies.

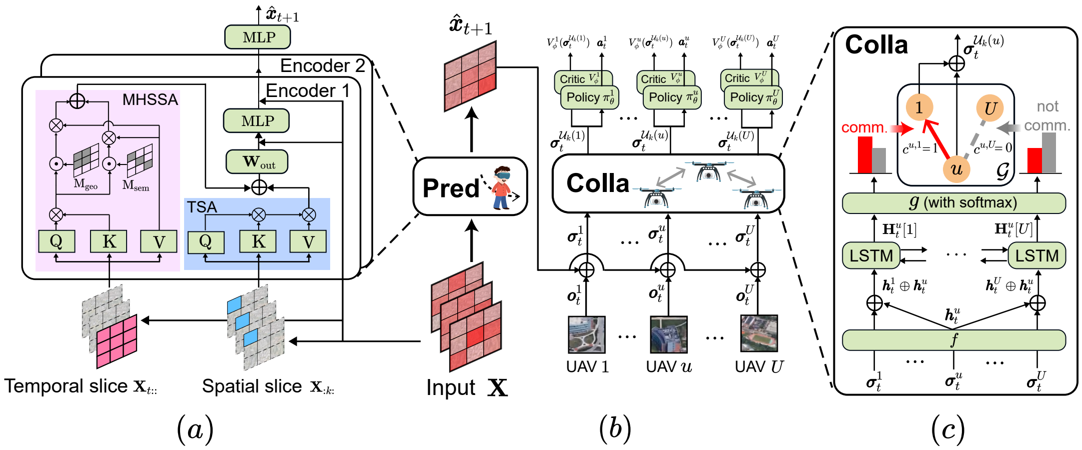

$\hat{\boldsymbol{x}}_{t+1}$
MLP
Encoder 2
Encoder 1
MHSSA
MLP
$\mathbf{W}_{\text{out}}$
M_{geo}
M_{sem}
TSA
Q
K
V
Q
K
V
Temporal slice $\mathbf{X}_{t::}$
Spatial slice $\mathbf{X}_{:k:}$
Input $\mathbf{X}$
$\hat{\boldsymbol{x}}_{t+1}$
Pred
$V_\phi^1(\boldsymbol{\sigma}_t^{\mathcal{U}_k(1)})$ $\boldsymbol{a}_t^1$
$V_\phi^u(\boldsymbol{\sigma}_t^{\mathcal{U}_k(u)})$ $\boldsymbol{a}_t^u$
$V_\phi^U(\boldsymbol{\sigma}_t^{\mathcal{U}_k(U)})$ $\boldsymbol{a}_t^U$
Critic V_ϕ^1
Policy π_θ^1
Critic V_ϕ^u
Policy π_θ^u
Critic V_ϕ^U
Policy π_θ^U
$\boldsymbol{\sigma}_t^{\mathcal{U}_k(1)}$
$\boldsymbol{\sigma}_t^{\mathcal{U}_k(u)}$
$\boldsymbol{\sigma}_t^{\mathcal{U}_k(U)}$
Colla
$\boldsymbol{\sigma}_t^1$
$\boldsymbol{\sigma}_t^u$
$\boldsymbol{\sigma}_t^U$
$\boldsymbol{o}_t^1$
$\boldsymbol{o}_t^u$
$\boldsymbol{o}_t^U$
UAV 1
UAV u
UAV U
Colla
$\boldsymbol{\sigma}_t^{\mathcal{U}_k(u)}$
1
U
comm.
not comm.
$c^{u,1}=1$
$c^{u,U}=0$
u
$\mathcal{G}$
g (with softmax)
$\mathbf{H}_t^u[1]$
$\mathbf{H}_t^u[U]$
LSTM
LSTM
$\boldsymbol{h}_t^1 \oplus \boldsymbol{h}_t^u$
$\boldsymbol{h}_t^U \oplus \boldsymbol{h}_t^u$
$\boldsymbol{h}_t^u$
f
$\boldsymbol{\sigma}_t^1$
$\boldsymbol{\sigma}_t^u$
$\boldsymbol{\sigma}_t^U$
(a)
(b)
(c)

Fig. 2: (a) Human-centric predictive framework "**Pred**", (b) Overall architecture for MetaCS, and (c) collaborative multi-UAV control framework "**Colla**".

1) Temporal Self-Attention (TSA): In metaverse applications, a mobile user's mobility pattern (e.g., periodic, trending) often dynamically changes over time, and thus satisfactory QoI experiences highly rely on the extraction and utilization of this temporal pattern. We employ TSA to model user behavior patterns in consecutive timeslots as a sequence. Specifically, we first obtain the query, key, and value matrices of self-attention operations for each region k as:

$$\mathbf{Q}_z^{(T)} = \mathbf{X}_{:z:}\mathbf{W}_Q^{(T)},\ \mathbf{K}_z^{(T)} = \mathbf{X}_{:z:}\mathbf{W}_K^{(T)},\ \mathbf{V}_z^{(T)} = \mathbf{X}_{:z:}\mathbf{W}_V^{(T)}, \tag{14}$$

where $\mathbf{W}_Q^{(T)}, \mathbf{W}_K^{(T)}, \mathbf{W}_V^{(T)} \in \mathbb{R}^{2\times d}$ are learnable matrixes and d is the embedding dimension. Then, we apply self-attention operations in the temporal dimension to model the temporal relationship of user behaviors between current timeslot t and previous L timeslots in z as:

$$\mathbf{A}_z^{(T)} = \frac{\mathbf{Q}_z^{(T)}(\mathbf{K}_z^{(T)})^\top}{\sqrt{d}} \in \mathbb{R}^{L\times L}. \tag{15}$$

Finally, we obtain the TSA output by multiplying the attention scores with the value matrix as:

$$\mathrm{TSA}(\mathbf{X}_{:z:}) = \mathrm{softmax}(\mathbf{A}_z^{(T)})\mathbf{V}_z^{(T)}. \tag{16}$$

2) Multi-Head Spatial Self-Attention (MHSSA): Similarly to TSA, we aim to capture the spatial relationship between different regions at current timeslot t, as:

$$\mathbf{Q}_t^{(S)} = \mathbf{X}_{t::}\mathbf{W}_Q^{(S)},\ \mathbf{K}_t^{(S)} = \mathbf{X}_{t::}\mathbf{W}_K^{(S)},\ \mathbf{V}_t^{(S)} = \mathbf{X}_{t::}\mathbf{W}_V^{(S)}, \tag{17}$$

$$\mathbf{A}_t^{(S)} = \frac{\mathbf{Q}_t^{(S)}(\mathbf{K}_t^{(S)})^\top}{\sqrt{d}} \in \mathbb{R}^{Z\times Z}, \tag{18}$$

where $\mathbf{A}_t^{(S)}$ is the attention score between any two regions in timeslot t. Then, we incorporate two heads to extract distinct spatial features, corresponding to two different masks: *geographic* mask $\mathbf{M}_{\text{geo}}$ and *semantic* mask $\mathbf{M}_{\text{sem}}$. $\mathbf{M}_{\text{geo}}$ is a binary matrix, whose weight is 1 if the distance between two regions is less than a threshold; and 0 otherwise. In this way, we can mask the attention of region pairs far away from each other and focus on nearby regions. On the other hand, we compute the similarity of past in/outflows between regions using the Dynamic Time Warping [43] algorithm. That is, we select N_{sem} regions with the highest similarity for each region as its semantic neighbors. Then, we construct the binary semantic masking matrix $\mathbf{M}_{\text{sem}}$ by setting the weight between the current node and its semantic neighbors to 1; and 0 otherwise. Thus we can find distant region pairs that exhibit similar behavior patterns. Then, MHSSA is calculated by:

$$\begin{aligned}\mathrm{MHSSA}(\mathbf{X}_{t::}) &= \mathrm{softmax}(\mathbf{A}_t^{(S)} \odot \mathbf{M}_{\text{geo}})\mathbf{V}_t^{(S)} \\ &\oplus \mathrm{softmax}(\mathbf{A}_t^{(S)} \odot \mathbf{M}_{\text{sem}})\mathbf{V}_t^{(S)},\end{aligned} \tag{19}$$

where $\odot$ indicates the Hadamard product and $\oplus$ is concatenate operation. In this way, the spatial self-attention module simultaneously incorporates geographic and semantic neighborhood information.

3) Spatial-Temporal Encoder: We concatenate TSA and MHSSA results, to integrate spatial and temporal information simultaneously, as:

$$\text{Encoder(X) = MLP}\Big(\big(\mathrm{TSA(X)}\oplus\mathrm{MHSSA}(\mathbf{X})\big)\mathbf{W}_{\text{out}} + \mathbf{X}\Big) + \mathbf{X}, \tag{20}$$

where $\mathbf{W}_{\text{out}}$ is a linear transformation. Note that here we add two residual connections [44] for securing training stability and define these operations as a Spatial-Temporal Encoder block. We adopt N_{block} blocks to accurately model the behaviors of mobile users and employ an MLP to output the final prediction result $\hat{\boldsymbol{x}}_{t+1}$. Then, **Pred** is updated by minimizing the mean squared error (MSE) between the true in/out flows and the predicted ones:

$$\mathcal{L}_{\text{pred}} = \mathbb{E}\left[(\hat{\boldsymbol{x}}_{t+1} - \boldsymbol{x}_{t+1})^2\right]. \tag{21}$$

Pred helps UAVs focus on those timeslots and location where network overload or service dropout will most likely happen, by capturing the spatio-temporal user movement pattern between regions. For each UAV u, we concatenate the predicted $\hat{\boldsymbol{x}}_{t+1}$ with its observation $\boldsymbol{o}_t^u$, to get a hybrid observation $\boldsymbol{\sigma}_t^u$. **Pred** can be easily extended to a multi-step prediction, by increasing the number of steps in the prediction process, at the cost of increased computational complexity.

To maintain low inference time (see Section VI-F), which is essential in our considered metaverse application scenario, we opt to utilize the one-step prediction in $\mathbf{Pred}$.

B. Collaborative Multi-UAV Navigations by Relational Graph-Guided IPPO

Having enriched the observation of all UAVs with predicted user movement in the next step is not enough to efficiently navigate UAVs to provide satisfactory QoI, this is because: (a) each UAV can only access the detailed information (i.e., the remaining data amount and data generation time) of the surrounding users within a fixed sensing range; and (b) each UAV can only obtain the current position or remaining energy of other UAVs through communications, which is not scalable in large environments where communication costs are prohibitively expensive. Therefore, we propose a collaborative multi-UAV navigation framework by using relational UAV communication graph-guided IPPO. However, UAV pairs may not affect with each other in certain scenarios (e.g., distant UAVs in charge of different regions). The softmax function assigns small but nonzero weights to unrelated and trivial UAVs, which weakens the benefits brought by a few interactive UAVs together.

Thus, we select an appropriate set of UAVs to communicate with, instead of learning the importance weight of all others. Formally, We abstract the relationship among UAVs as a directed graph $\mathcal{G} = (\mathcal{U}, \mathcal{C})$, where each node u corresponds to a UAV, and the property of node u is the hybrid observation $\boldsymbol{\sigma}_t^u$ received by a UAV u at t. The property of directed edge $c_t^{u,u'} \in \{0, 1\}$ acts as an indicator, suggesting whether a UAV u communicates with a UAV u' or not (i.e., taking $\boldsymbol{\sigma}_t^{u'}$ of a UAV u' into considerations) at t. Note that $c_t^{u,u'}$ is not equivalent to $c_t^{u',u}$. We denote $\boldsymbol{c}_t^u := \left[c_t^{u,1}, \cdots, c_t^{u,U}\right]$ as a vector containing all directed edges starting from a node u, as:

$$\boldsymbol{h}_t^u = f(\boldsymbol{\sigma}_t^u), \tag{22}$$

$$\mathbf{H}_t^u = \text{Bi-LSTM}\left[\left(\boldsymbol{h}_t^u \oplus \boldsymbol{h}_t^1, \cdots, \boldsymbol{h}_t^u \oplus \boldsymbol{h}_t^U\right)\right], \tag{23}$$

$$\boldsymbol{c}_t^u = g(\mathbf{H}_t^u), \tag{24}$$

where $\boldsymbol{\sigma}_t^u$ is encoded into a feature $\boldsymbol{h}_t^u$ by an MLP function $f(\cdot)$. Then, $\boldsymbol{h}_t^u$ is concatenated with the feature of all UAVs, to form the input of Bi-LSTM whose sequence length is U, and we obtain the embedding matrix $\mathbf{H}_t^u \in \mathbb{R}^{U \times d}$, where d is the embedding dimension. In contrast to LSTM whose output only depends on the feature of the current and the previous UAV but ignores all other UAVs, Bi-LSTM is able to incorporate $\boldsymbol{\sigma}_t^u$ of all UAVs. Inspired by the gating mechanism of IC3Net [45], we use an MLP function $g(\cdot)$ with softmax layer for two choices (i.e., whether to communicate or not) to obtain $\boldsymbol{c}_t^u$. Note that each row of $\mathbf{H}_t^u$ are fed into $g(\cdot)$ in parallel.

Algorithm 1 MetaCS

```
Initialize parameters of $\mathbf{Pred}$, $\mathbf{Colla}$; Initialize policy network $\pi_\theta^u$, value network $V_\phi^u$ for each UAV $u$;
for iteration= $1, 2, \cdots$ do
    for $t = 0, 2, \cdots, T-1$ do
        Calculate predicted user flow $\hat{\boldsymbol{x}}_{t+1} = \mathbf{Pred}(\mathbf{X})$;
        for $u = 1, 2, \cdots, U$ do
            Get hybrid observation $\boldsymbol{\sigma}_t^u = \boldsymbol{o}_t^u \oplus \hat{\boldsymbol{x}}_{t+1}$;
            Calculate graph-guided feature $\boldsymbol{\sigma}_t^{\mathcal{U}_k(u)} = \mathbf{Colla}(\boldsymbol{\sigma}_t^u, \{\boldsymbol{\sigma}_t^{u'}\}_{u' \neq u})$;
            Select action $\boldsymbol{a}_t^u \sim \pi_\theta^u\left(\boldsymbol{\sigma}_t^{\mathcal{U}_k(u)}\right)$;
        end for
        Interact the environment with $\{\boldsymbol{a}_t^u\}_{u=1}^U$;
    end for
    Update $\mathbf{Pred}$ by Eqn. (21);
    Update $\pi_\theta^u$, $V_\phi^u$, $\mathbf{Pred}$, $\mathbf{Colla}$ by Eqn. (27) and Eqn. (28);
end for
```

To better optimize the UAV policies, we need to exploit the learned relational graph $\mathcal{G}$. We define the set of k-hop neighbors of a UAV u as $\mathcal{U}_k(u)$, where $u' \in \mathcal{U}_k(u)$ indicates there exists a directed path from u to u' with length at most k in graph $\mathcal{G}$. Based on $\mathcal{U}_k(u)$, each UAV u selects the most appropriate set of UAVs to communicate with and generate a compact graph-guided feature $\boldsymbol{\sigma}_t^{\mathcal{U}_k(u)}$ as:

$$\boldsymbol{\sigma}_t^{\mathcal{U}_k(u)} := \{\oplus \boldsymbol{\sigma}_t^{u'} | \forall u' \in \mathcal{U}_k(u)\}. \tag{25}$$

Typically, higher values of k tends to yield better overall performance, but it also incurs greater computational costs and increase the difficulty of policy learning. We will treat k as a hyperparameter and show the tuning results in Section VI-B.

After selecting informative UAVs for dynamic interactions by $\mathbf{Colla}$, UAVs are capable of executing efficient graph-guided decentralized control. This is because that on one hand, it is able to avoid unnecessary and costly communication between UAVs responsible for different regions to ensure QoI for metaverse. On the other hand, via UAV communication with selected interactive partners, graph-guided IPPO is able to mitigate the non-stationary problem of general MADRL and thus improving the attained overall QoI index, especially in situations when network overload or service dropout happens. Therefore, effective UAV collaborations is crucial. Formally, based on the naive IPPO, policy and value function for each UAV u are parameterized by θ and ϕ, represented as π_θ^u and V_ϕ^u, respectively. By using $\mathbf{Colla}$, the graph-guided value function is:

$$V_\phi^u\left(\boldsymbol{\sigma}_t^{\mathcal{U}_k(u)}\right) = \mathbb{E}_{\pi_\theta^u}\left[\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'}^u \middle| \boldsymbol{\sigma}_t^{\mathcal{U}_k(u)}\right], \quad \forall u, \tag{26}$$

and updated by minimizing the temporal-difference error [46], as:

$$\mathcal{L}_{\text{value}}^u = \mathbb{E}_{\pi_\theta^u}\left[r_t^u + \gamma V_\phi^u\left(\boldsymbol{\sigma}_{t+1}^{\mathcal{U}_k(u)}\right) - V_\phi^u\left(\boldsymbol{\sigma}_t^{\mathcal{U}_k(u)}\right)\right]^2, \quad \forall u. \tag{27}$$

Then a UAV's policy π^u_θ is updated by using the importance sampling weight:

$$\mathcal{L}^u_{\text{policy}} = \mathbb{E}_{\pi^u_\theta}\Bigg[\min\Bigg(\frac{\pi^u_\theta\left(\boldsymbol{a}^u_t\middle|\boldsymbol{\sigma}^{\mathcal{U}_k(u)}_t\right)}{\pi^u_{\text{old}}\left(\boldsymbol{a}^u_t\middle|\boldsymbol{\sigma}^{\mathcal{U}_k(u)}_t\right)}A^u_t, \\ \text{clip}\left(\frac{\pi^u_\theta\left(\boldsymbol{a}^u_t\middle|\boldsymbol{\sigma}^{\mathcal{U}_k(u)}_t\right)}{\pi^u_{\text{old}}\left(\boldsymbol{a}^u_t\middle|\boldsymbol{\sigma}^{\mathcal{U}_k(u)}_t\right)}, 1-\epsilon, 1+\epsilon\right)A^u_t\Bigg)\Bigg], \forall u, \tag{28}$$

where $A^u_t = r^u_t + \gamma V^u_\phi\left(\boldsymbol{\sigma}^{\mathcal{U}_k(u)}_{t+1}\right) - V^u_\phi\left(\boldsymbol{\sigma}^{\mathcal{U}_k(u)}_t\right)$ is the advantage function, and ϵ is a hyper-parameter that controls the learning stability.

C. Algorithm Details

The pseudocode of MetaCS is given in Algorithm 1. First, we utilize a Xavier uniform initializer [47] for all learnable parameters (Line 1). In each timeslot t, we calculate the predicted user flow $\hat{\boldsymbol{x}}_{t+1}$ through $\mathbf{Pred}$ (Line 4). Then, each UAV u gets a hybrid observation $\boldsymbol{\sigma}^u_t$ by merging local observation $\boldsymbol{o}^u_t$ with the predicted user flow (Line 6), and utilizes $\mathbf{Colla}$ to select the most appropriate set of UAVs to communicate with and generate a compact graph-guided feature $\boldsymbol{\sigma}^{\mathcal{U}_k(u)}_t$ (Line 7). At last, a UAV u selects an action $\boldsymbol{a}^u_t$ according to the current policy network π^u_θ (Line 8). $\{\boldsymbol{a}^u_t\}^U_{u=1}$ is used to interact with the environment and each UAV u receives the reward r^u_t (Line 10). During each iteration, we first optimize $\mathbf{Pred}$ for a more accurate user flow prediction (Line 12). Then, π^u_θ, V^u_ϕ, $\mathbf{Pred}$, $\mathbf{Colla}$ are optimized several times as sample reuse in IPPO [19], by minimizing $\mathcal{L}^u_{\text{value}}$ and maximizing $\mathcal{L}^u_{\text{policy}}$ (Line 13).

Apart from IPPO, MetaCS can also work with other multi-agent actor-critic frameworks, such as MAPPO [31] and IA2C [48]. For MAPPO, the input of critic networks V^u_ϕ is replaced from $\sigma^{\mathcal{U}_k(u)}_t$ to the global state s_t; see Eqn. (26). For IA2C, the loss function of UAV's policy is modified as $\mathcal{L}^u_{\text{policy}} = \mathbb{E}_{\pi^u_\theta}\left[A_t \log \pi^u_\theta(\boldsymbol{a}^u_t|\boldsymbol{\sigma}^{\mathcal{U}_k(u)}_t)\right], \forall u$; see Eqn. (28).

VI. Experimental Results

A. Dataset Descriptions and Experimental Settings

We conduct extensive experiments on both campus-scale and urban-scale task regions. For the former, we utilize datasets from NCSU [2] (with an area of $3.3 \times 2.8\text{km}^2$ and 35 students) and KAIST (with an area of $2.1 \times 2.2\text{km}^2$ and 92 students), where student movement traces are recorded by GPS receivers. One typical metaverse application in campus-scale environments is AR tourism (since as the size of tourist attractions is usually comparable to that of campuses), where virtual elements are overlaid onto real-world environments to enhance a visitor's experience, like viewing virtual reconstructions of ancient ruin through AR glasses. For the latter, we use a dataset from Beijing [49] comprising 150 taxi movement traces within the city's downtown region as an area of $6.5 \times 4.0\text{km}^2$. One typical metaverse application in urban-scale environments is AR navigations, which offers real-time intuitive guidance to drivers by presenting virtual arrows and street names.

TABLE II: Simulation Settings (unchanged parameters)

Notation	Value	Notation	Value	Notation	Value
T	120	δ	20s	P	10
v	20m/s	D	60Mbits	w_1	4.88
w_2	0.43	w_{LoS}	1	w_{NLoS}	20
B	240MHz	e_0	719.2KJ	ρ_0	0.01w
β_0	-30dBm	F	0.94	f	2GHz

Simulation settings are summarized in Table II. For MU-MIMO, we referenced the parameter settings in [35]; energy supply and flight capabilities of UAVs were taking from the public report of DJI Matrice300 RTK[3]. By default, we consider that $U = 5$, $U = 5$, $U = 10$ UAVs are deployed, set $T = 120$, $T = 120$, $T = 240$, in NCSU, KAIST and Beijing respectively. We set AoI threshold $\tau^m_{\text{th}} = 30$ timeslots and throughput requirement $\mu^m_{\text{th}} = 3$ Mbps for all users at each timeslot. We employed PyTorch as the implementation framework and trained all models on Ubuntu 20.04.5 LTS with GeForce RTX A6000 GPUs. We conducted a comprehensive set of experiments, including hyperparameter tuning, ablation study, performance comparison with six baselines, and trajectory visualization. Results were assessed using the overall QoI index I, as well as individual ones including episodic AoI τ, throughput satisfactory ratio I_μ, AoI satisfactory ratio I_τ, and average energy consumption $\bar{e}$.

B. Hyperparameters Tuning

We select two key parameters: (a) the number of regions $Z \in \{9, 36, 81, 144\}$ in user modeling to study the impact of region partition granularity, and (b) the number of hop $k \in \{0, 1, 2\}$ to investigate the impact of UAV communications. The embedding dimension $d = 64$; number of neighbors in MHSSA $N_{\text{sem}} = 5$; sequence length $L = 5$ and $N_{\text{block}} = 2$ blocks in $\mathbf{Pred}$. For remaining hyper-parameters, we adopt the commonly used settings in [31], which include using the Adam optimizer for all learnable parameters and setting $\gamma = 0.99$. Results are given in Table III. We observe the following. Overall QoI index reaches a peak 3.190 when $Z = 36$ in NCSU, and 3.138 when $Z = 81$ in KAIST. This is because smaller Z may lead to a reduction of user in/outflow density, however larger values may increase the computational complexity of learning. Due to the difference of workzone size and number of students between NCSU and KAIST, we set the optimal granularity as $Z = 36$ and $Z = 81$ for two campuses, respectively. On the other hand, when k increases, we see that overall QoI index first goes up and then decreases in both datasets. For example, MetaCS achieves $I = 2.977$ and 2.939 for $k = 0$, and $I = 3.078$ and 3.123 for $k = 2$, which is worse than the attained I for $k = 1$. In general, larger k leads to better results on UAV trajectory planning but also higher inter-UAV communication costs. We see a peak value due to possible redundant UAV interactions, which increases the solution space and does harm to the stability of IPPO. When UAVs are deployed in a very large-scale workzone, they do not require global knowledge to make individual decisions, thus it is more appropriate for them to receive only

[2] https://crawdad.org/ncsu/mobilitymodels/20090723

[3] https://www.dji.com/cn/matrice-300

TABLE III: Hyperparameters Tunning

Dataset		NCSU					KAIST				
Metric		I	I_τ	I_μ	τ	$\bar{e}$	I	I_τ	I_μ	τ	$\bar{e}$
$k = 0$	$Z = 9$	2.950	0.894	0.903	12.581	0.303	2.855	0.893	0.871	12.998	0.305
	$Z = 36$	2.977	0.901	0.919	11.832	0.303	2.860	0.877	0.893	11.916	0.307
	$Z = 81$	3.017	0.961	0.915	10.001	0.302	2.939	0.907	0.897	11.331	0.305
	$Z = 144$	2.935	0.890	0.915	12.218	0.303	2.863	0.878	0.884	11.457	0.307
$\boldsymbol{k = 1}$	$Z = 9$	3.125	0.970	0.944	9.215	0.302	2.998	0.928	0.919	9.239	0.307
	$\boldsymbol{Z = 36}$	**3.190**	**0.981**	**0.968**	**6.676**	**0.303**	3.086	0.957	0.943	9.326	0.305
	$\boldsymbol{Z = 81}$	3.118	0.968	0.943	8.801	0.302	**3.138**	**0.960**	**0.961**	**8.288**	**0.306**
	$Z = 144$	3.069	0.957	0.928	9.747	0.302	3.056	0.191	0.932	9.653	0.305
$k = 2$	$Z = 9$	3.054	0.972	0.923	9.909	0.302	3.080	0.189	0.939	10.009	0.305
	$Z = 36$	3.078	0.971	0.931	8.203	0.303	3.106	0.194	0.947	8.169	0.305
	$Z = 81$	3.117	0.975	0.943	8.541	0.303	3.123	0.194	0.956	7.364	0.306
	$Z = 144$	3.071	0.953	0.928	10.239	0.302	3.083	0.192	0.941	8.628	0.305

adjacent information from neighbors. The above best set of hyperparameters will be used in the subsequent experiments.

C. Ablation Study

We perform ablation study by gradually removing two key components of MetaCS. Based on the base design of IPPO, from Table IV, we see that the complete MetaCS yields 8.9% and 8.2% improvements on I_τ and I, respectively, compared to MetaCS w/o $\mathbf{Pred}$ in NCSU dataset. Due to more dispersed users and higher degree of unpredictability in NCSU (if compared to KAIST), $\mathbf{Pred}$ is required to obtain much more precise in/outflow user modeling. In contrast to NCSU, there are more users in KAIST and many of them tend to gather together during certain periods in core areas like dormitories, dining halls, and classrooms. The presence of large crowd results in more severe occurrence of possible network overload if insufficient number of UAVs are scheduled over. Benefited from our proposed efficient UAV coordination mechanism, MetaCS achieves $6.4\%, 6.7\%$ improvement of I_μ, I compared to that when removing $\mathbf{Colla}$ in KAIST, respectively. Furthermore, when both $\mathbf{Pred}$ and $\mathbf{Colla}$ are removed, we observe obvious overall QoI index drop, demonstrating the effectiveness of combining two together. Note that although the total moving distance of UAVs is increased when using $\mathbf{Pred}$ and $\mathbf{Colla}$, the energy consumption has barely changed, since the consumed energy when flying at maximum speed (3369KJ) is comparable to hovering (3180KJ), based on the energy model of UAVs given by Section III. Thus, MetaCS enhance the QoI while maintaining nearly constant energy consumption.

Based on the base design of IA2C [48], we find that complete MetaCS can also achieve superior performance. However, without our IPPO's reuse technique, IA2C typically exhibits lower sample efficiency, as a result of lower attained overall QoI index I. Therefore, IPPO is chosen as the basis for MetaCS.

D. Performance Comparison with Baselines

We compare MetaCS with six baselines:

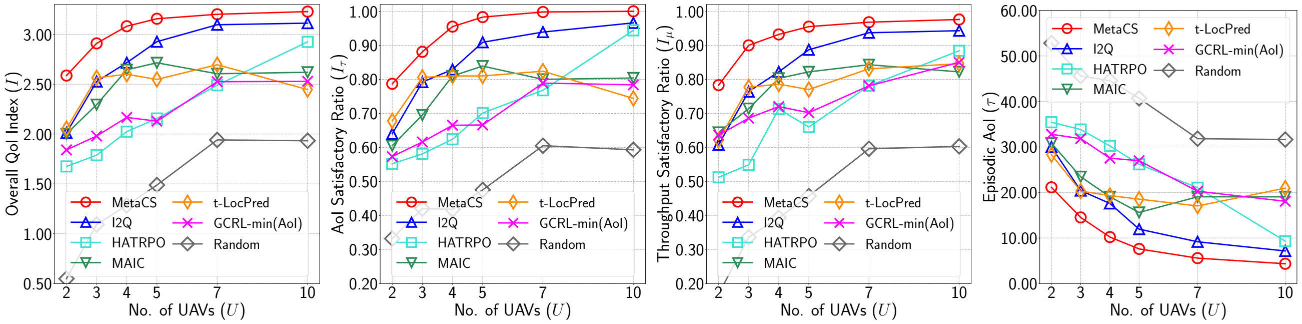
- I2Q [50]: It learns an independent actor and critic for each agent, by modeling the optimal conditional joint policy of other agents without any communication between them. We consider it as the state-of-the-art decentralized MADRL approach.
- HATRPO [51]: It employs a centralized critic and a policy iteration procedure with monotonic improvement guarantee, which is considered as an enhanced version of MAPPO [31] and state-of-the-art centralized MADRL approach. We use it to show the benefits of our graph-guided UAV navigation compared with sharing all information.
- MAIC [52]: It proposes a multi-agent incentive communication (MAIC) framework, which enables agents to exchange information based on teammate modeling for explicit coordination. It is considered as the state-of-the-art communication-based MADRL approach and we use it to validate the benefits of our graph-guided UAV collaboration.
- t-LocPred [53]: It employs a memory-augmented attentive sequential model to deal with the difficulty of long-term user modeling, which is considered as the state-of-the-art user behavior modeling approach. We integrate it with the IPPO backbone, in a manner consistent with MetaCS.
- GCRL-minAoI [33]: It is the state-of-the-art DRL approach to minimize AoI of all mobile users in MCS, which consists of a model-based MCTS [18] structure for path planning and a relational graph convolutional network to extract spatial correlation between UAVs and mobile users.
- Random: Each UAV u chooses an action $\boldsymbol{a}_t^u$ randomly from action space $\mathcal{A}$.

To justify the effectiveness and robustness of MetaCS, we vary the number of UAVs, throughput requirement and AoI threshold, respectively. We explicitly investigate tasks with extensive VR requests by increasing the throughput requirement, and tasks with extensive AR requests by decreasing the AoI threshold.

1) Impact of the number of UAVs U: We vary $U \in \{2, 3, 4, 5, 7, 10\}$ in NCSU and KAIST datasets, and from Fig. 3 and Fig. 4, we observe that MetaCS consistently outperforms all six baselines in terms of overall QoI index I. With fewer UAVs, the benefits offered by $\mathbf{Pred}$ and $\mathbf{Colla}$ modules are crucial to achieve the desirable QoI level. Taking NCSU for example, since it is a big campus, only deploying two UAVs is far from sufficient, however MetaCS still obtains $I = 2.587$ which is 29% higher than the best baseline I2Q. Furthermore, we can see that the capability for UAVs to accomplish tasks

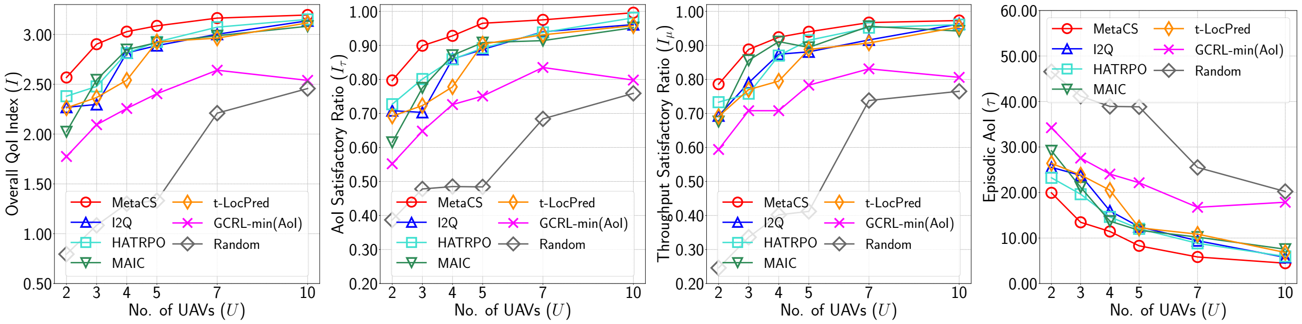
TABLE IV: Ablation Study

Base designs	Improvements	NCSU					KAIST				
		I	I_τ	I_μ	τ	$\bar{e}$	I	I_τ	I_μ	τ	$\bar{e}$
IPPO	MetaCS	**3.190**	**0.981**	**0.968**	**6.676**	**0.303**	**3.138**	**0.960**	**0.961**	**8.288**	**0.306**
	MetaCS w/o **Pred**	2.947	0.892	0.912	12.264	0.303	3.070	0.947	0.937	9.299	0.305
	MetaCS w/o **Colla**	2.977	0.901	0.919	11.832	0.303	2.939	0.907	0.897	11.331	0.305
	MetaCS w/o **Pred** & **Colla**	2.899	0.877	0.900	14.135	0.302	2.855	0.902	0.872	11.944	0.305
IA2C	MetaCS	**2.999**	**0.927**	**0.908**	**12.788**	**0.303**	**3.072**	**0.942**	**0.943**	**8.315**	**0.306**
	MetaCS w/o **Pred**	2.748	0.885	0.832	16.106	0.303	2.921	0.894	0.896	13.169	0.306
	MetaCS w/o **Colla**	2.974	0.907	0.900	14.717	0.302	2.571	0.814	0.790	17.913	0.306
	MetaCS w/o **Pred** & **Colla**	2.724	0.863	0.825	17.194	0.302	2.564	0.797	0.786	18.758	0.306



(a) I (b) I_τ (c) I_μ (d) τ

Fig. 3: Impact of the number of UAVs U (NCSU).

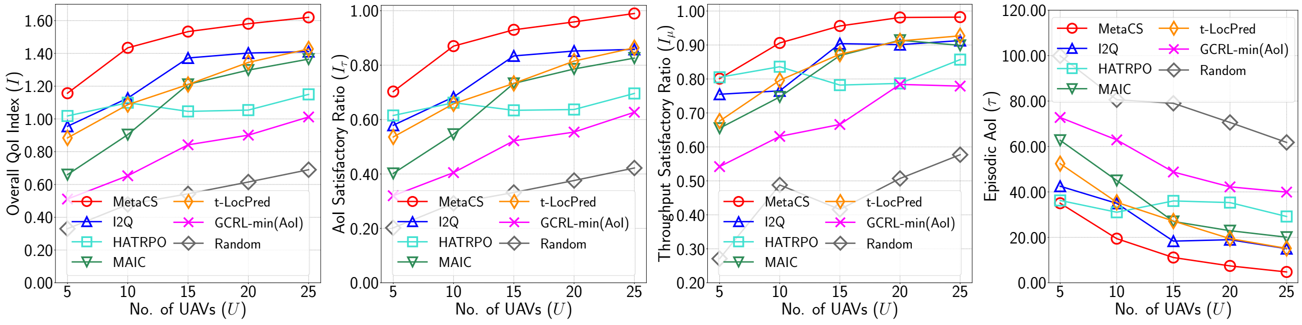


(a) I (b) I_τ (c) I_μ (d) τ

Fig. 4: Impact of the number of UAVs U (KAIST).

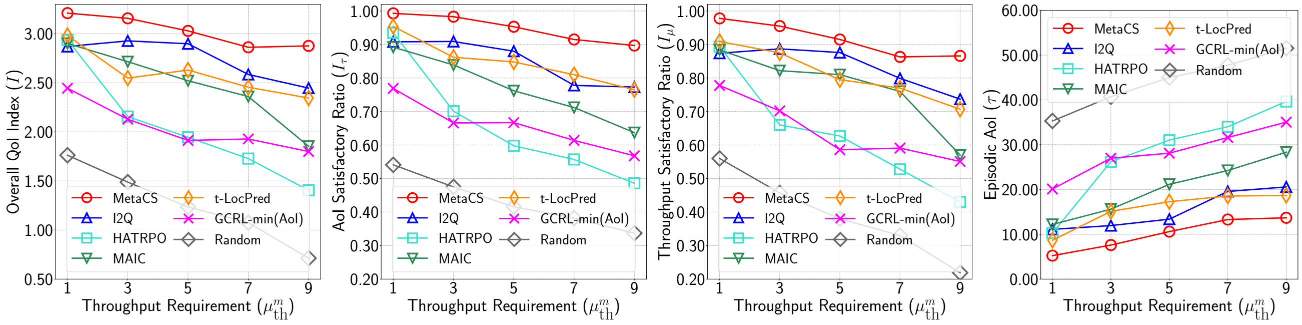
in metaverse has been gradually improved as U increases. In NCSU with relatively lower user density, MetaCS is sufficient to achieve nearly 100% throughput satisfactory ratio and AoI satisfactory ratio with 10 deployed UAVs. However, when excessive UAVs are used, the performance improvement of GCRL-min(AoI) becomes marginal and may even lead to a decline in overall QoI index (see Fig. 4). This is because it suffers from the exponential increased tree-search space w.r.t U. The attained QoI by all methods in Beijing dataset is half of that in campus environments such as NCSU and KAIST, due to the fact that its task duration and energy consumption of each UAV being doubled. However, MetaCS still achieve the highest QoI when varying the number of UAV from 5 to 25, as shown in Fig. 5, which demonstrates its capability to scale to larger task region involving more users and deployed UAVs.

2) Impact of throughput requirement μ_{th}^m: We vary μ_{th}^m from 1Mbps to 9Mbps for all users. From Fig. 6, Fig. 7 and Fig. 8, we see that the overall QoI index I by MetaCS consistently exceeds that of six baselines. For example, when $\mu_{\text{th}}^m = 9$Mbps, MetaCS achieves the highest $I = 2.876$ in NCSU, which is $17.6\%, 18.3\%$ higher than that of I2Q and t-LocPred, respectively. The best baseline I2Q performs worse when μ_{th}^m is increased from 7Mbps to 9Mbps in NCSU. This is because the employed decentralized method is unable to construct effective UAV collaboration when users request high throughput demand, such as immersive VR games. In this case, navigating UAVs closer to each other as a group is crucial to increase the provided data rate by uplink MU-MIMO. Furthermore, when μ_{th}^m increases, I_μ of most methods decreases rapidly, leading to a decline in overall I. However, MetaCS still remains at a high level, attributed to the fact that it sacrifices some degree of I_τ (e.g., by giving up data collections for a small number of remote users) in exchange for a less drastic drop in I_μ, since overall QoI index evaluates their bottleneck. We observe that HATRPO receives significantly lower I in NCSU than in KAIST, which also drops drastically with higher μ_{th}^m. The reason is that HATRPO lacks the user modeling mechanism, which makes it unable to accurately capture their movement patterns between regions in the NCSU



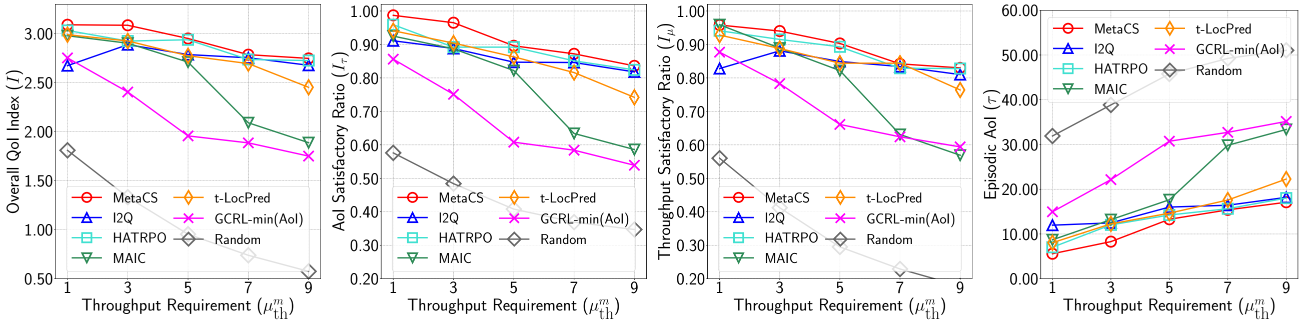
(a) I (b) I_τ (c) I_μ (d) τ

Fig. 5: Impact of the number of UAVs U (Beijing).



(a) I (b) I_τ (c) I_μ (d) τ

Fig. 6: Impact of throughput requirement μ_{th}^m (NCSU).

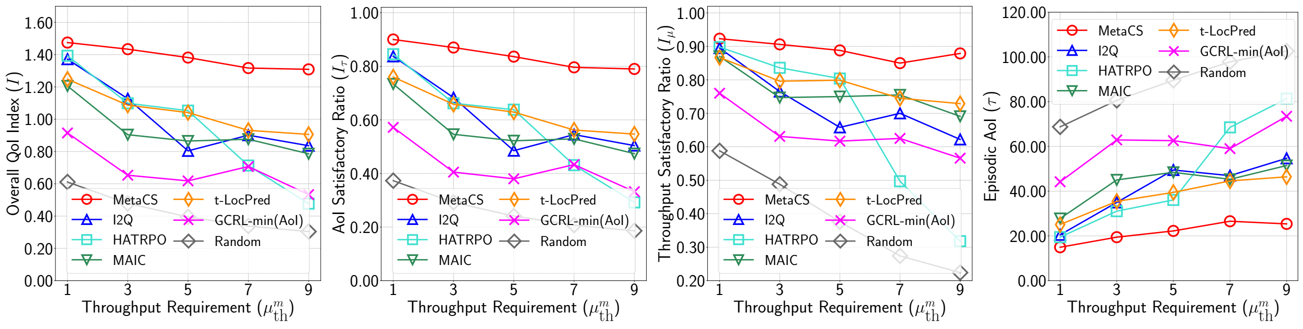


(a) I (b) I_τ (c) I_μ (d) τ

Fig. 7: Impact of throughput requirement μ_{th}^m (KAIST).

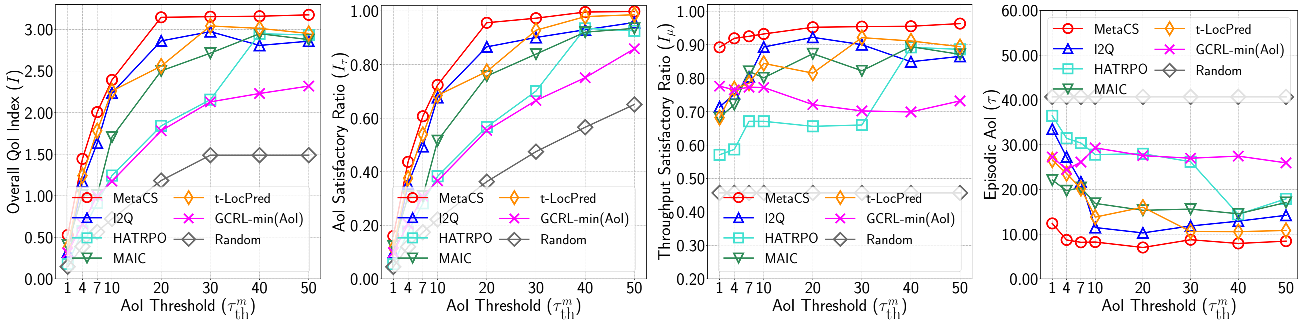
workzone, that when greater in/outflow dynamics happen, UAVs cannot be properly scheduled for services. Also, since HATRPO is a fully centralized approach, the local policy space of UAVs is expanded, thereby increasing the training complexity which results in sub-optimal UAV coordination in the complex NCSU environment. Finally, MetaCS outperforms all other baselines in Beijing dataset, since taxis exhibits considerable mobility increase compared to students walking in campuses, thus the spatiotemporal distribution of users (i.e., taxis) is more complex and unpredictable, making accurate user prediction crucial.

3) Impact of AoI threshold τ_{th}^m: Fig. 9, Fig. 10 and Fig. 11 show the impact of application determined AoI threshold τ_{th}^m of all users by varying it from 1 to 50 timeslots. When the τ_{th}^m varies from 20 to 50 timeslots, MetaCS yields QoI index I significantly surpasses all six baselines, especially in NCSU. For example, when $\tau_{\text{th}}^m = 20$ timeslots, MetaCS achieves the highest overall QoI index $I = 3.146$ in NCSU, as $9\%, 22\%$ higher than two best baselines I2Q and t-LocPred, respectively. We further observe that I obtained by MetaCS almost remains constant in these settings, since it is able to optimize the AoI satisfactory ratio to nearly 100%, and then throughput satisfaction ratio becomes the bottleneck. When decreasing τ_{th}^m from 30 to 10 timeslots (i.e., users request for more delay-sensitive metaverse services), all methods suffer from a significant decrease in I_τ, however MetaCS still achieves highest degree of QoI satisfaction. We also see a significant decline in I obtained by MAIC, because its attention-like communication allocates non-zero weights to insignificant and irrelevant UAVs, thereby diminishing the advantages conferred by UAV interactions. Furthermore, when τ_{th}^m is decreased from 10 to 1 timeslot (which corresponds to the applications that needs AR collaborations of stringent latency requirements), MetaCS still outperforms all other baselines in terms of overall



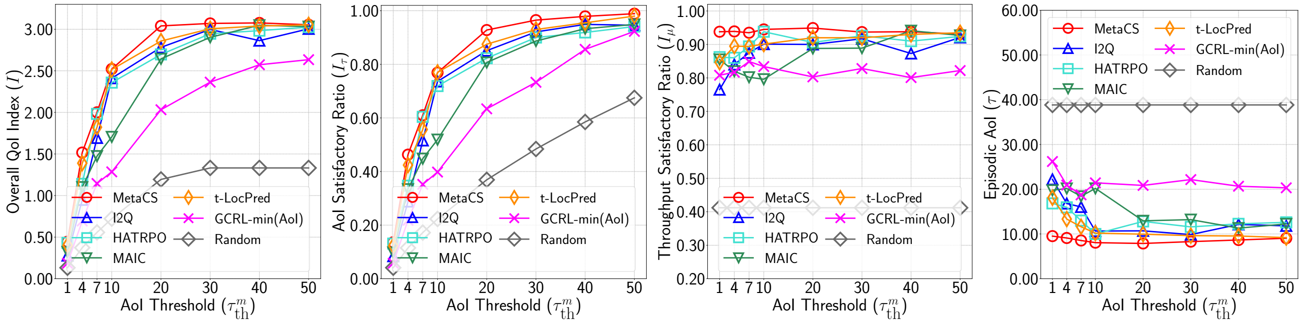
(a) I (b) I_τ (c) I_μ (d) τ

Fig. 8: Impact of throughput requirement μ_{th}^m (Beijing).



(a) I (b) I_τ (c) I_μ (d) τ

Fig. 9: Impact of AoI threshold τ_{th}^m (NCSU).



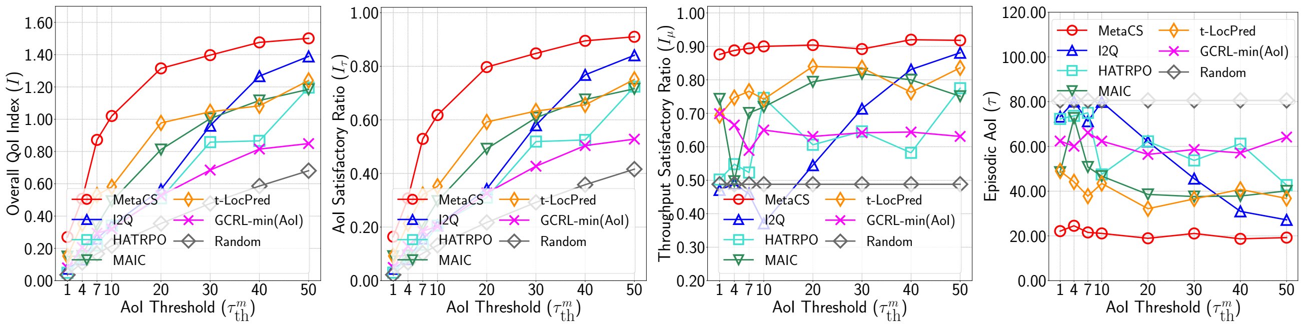
(a) I (b) I_τ (c) I_μ (d) τ

Fig. 10: Impact of AoI threshold τ_{th}^m (KAIST).

QoI index. When we decrease AoI threshold τ_{th}^m, MetaCS and t-locPred are better suited to cope with the possible server dropout, compared to other baselines which do not well model user behaviors. Our designed **Pred** module explicitly takes both geographic and semantic neighbors into consideration that assists to achieve a 6% improvement over t-locPred when $\tau_{\text{th}}^m = 20$ timeslots in KAIST dataset. Finally, MetaCS attains highest overall performance in terms of QoI index in Beijing dataset. With larger task region and more involved users, UAVs need to accomplish a dynamic spatial division of labor through appropriate collaborations. We find that the throughput satisfactory ratio achieved by all methods except random fluctuates with different AoI thresholds, since the AoI satisfaction becomes the performance bottleneck.

E. Trajectory Visualizations

For clearer demonstrations, we show the trajectories of UAVs given by MetaCS and attained metrics are displayed in the form of heatmap. $U = 3$, $U = 3$ and $U = 10$ UAVs are deployed in NCSU, KAIST and Beijing, respectively. In NCSU (as shown in Fig. 12), its core area hosts larger number of user regular movements, and thus requiring more UAVs to fly over. Meanwhile, there are also many other users located in remote areas (e.g., upper left, upper right, and lower left regions), and ignoring them completely would harm the task-level overall QoI index. We observe that the green and blue UAVs move back and forth between the core area and remote areas, but not in the same pace but in different timeslots. This indicates that they learn to well complement each other to ensure that users are always being serviced by enough UAVs. For KAIST (Fig. 13), even more users are concentrated



(a) I (b) I_τ (c) I_μ (d) τ

Fig. 11: Impact of AoI threshold τ^m_{th} (Beijing).

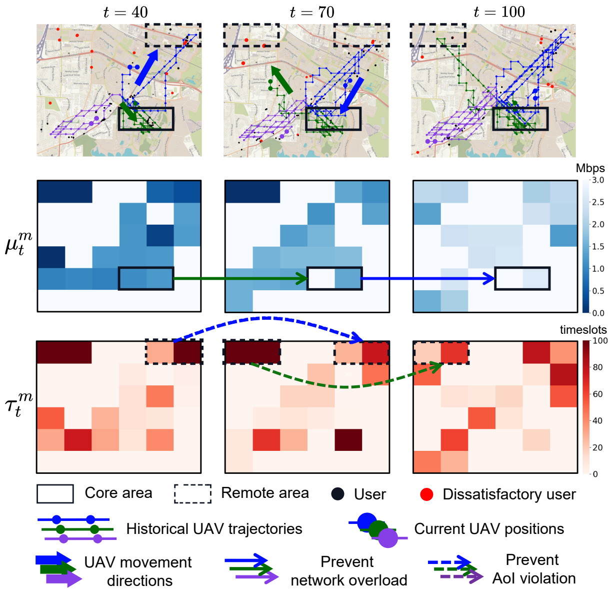


Fig. 12: Visualized trajectories of UAVs and attained metrics in the form of heatmap (NCSU). Each column corresponds to a particular timeslot; Heapmaps in the second row show the average μ^m_t of all users within a region, to indicate whether the data upload requests of most users in the core area are being met sufficiently or not. Heatmaps in the third row show the sum of episodic AoI τ^m_t of all users within a region, to intuitively indicate which areas are experiencing more AoI violation.

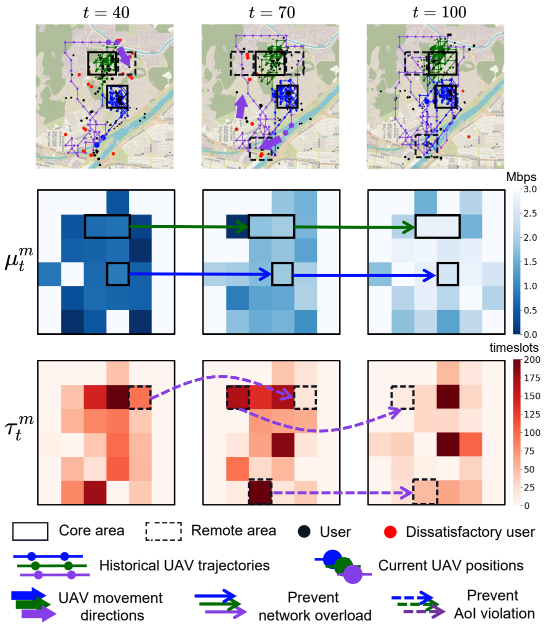


Fig. 13: Visualized trajectories of UAVs and attained metrics (KAIST).

in the central part of the campus, leaving fewer users in remote regions. Therefore, we observe that green and blue UAVs are servicing the concentrated users almost at all times, to avoid possible network overload. Meanwhile, the purple UAV tends to move across different remote regions to cover corner cases. In Beijing dataset (Fig. 14), the eastern part, known as the central business district, usually accommodates very high density of taxis. Thus we see that one single UAV (purple) is responsible for covering the western part to prevent AoI violation. Meanwhile, the remaining nine UAVs (blue) generate looped trajectories exclusively within the eastern part, thereby achieving a clear spatial division of labor to support satisfactory network throughput for metaverse applications.

The above benefits in all campuses are achieved by our proposed user behavior prediction and UAV collaboration modules, where precise prediction of crowd flows enables UAVs to access remote areas just in time, while accurate UAV interactions ensures that spatial-temporal throughput requirement is guaranteed.

To further confirm this, we use the root mean square error (RMSE) to evaluate the accuracy of **Pred** in MetaCS, denoted as $\sqrt{\frac{1}{T}\sum_{t=0}^{T-1}(\boldsymbol{x}_{t+1}-\hat{\boldsymbol{x}}_{t+1})^2}$. From Fig. 15(a), we observe that the RMSE obtained by MetaCS are 0.324, 0.263 and 0.404 on three datasets, which are lower than the ones by DeepCrowd [54], and significantly lower than t-LocPred as 0.529, 0.556 and 0.450, showing a reduction of 38.8%, 52.7% and 10.2% of RMSE, respectively. This is because MetaCS has the comprehensive ability to capture both temporal and spatial dependencies through TSA and MHSSA. Furthermore, MetaCS employs both the geographic and semantic masks, which explicitly considers long-range spatial dependencies in the regions where similar movement patterns may be dispersed throughout different locations, rather than being clustered together. These techniques are not considered in either the

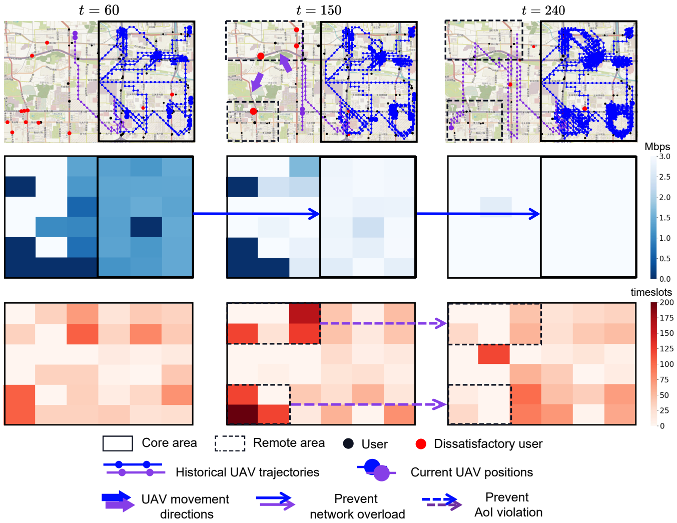
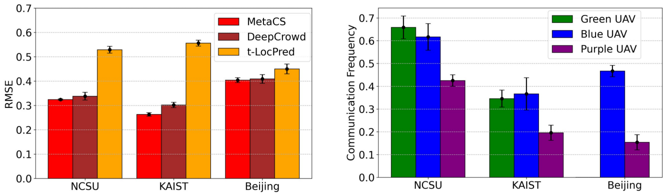


Fig. 14: Visualized trajectories of UAVs and attained metrics (Beijing).



(*a*) (*b*)

Fig. 15: (a) Comparison of user flow prediction performance between MetaCS, DeepCrowd, and t-LocPred. (b) Communication frequency among UAVs in **Colla**. Here error bars represent the standard deviation over 5 runs.

CNN-based t-LocPred or DeepCrowd which are built on the pyramid ConvLSTM architecture, and thus MetaCS exhibits better flow prediction performance.

Fig. 15(b) illustrates the communication frequency among UAVs in **Colla**, as the the frequency of a particular UAV u incorporates $\boldsymbol{\sigma}_t^{u'}$ of other UAVs $\{u'\}$ during the entire task. We see that UAV interactions are frequent in NCSU and Beijing datasets, evidenced by the communication frequency of the green UAV in NCSU and the blue UAVs in NCSU and Beijing datasets, respectively. Due to the larger task region in the these two datasets, inter-UAV communication can assist each UAV to acquire more information where AoI violations or throughput dissatisfaction happen in certain regions. As a result, they fly over to provide emergent services. In contrast, the purple UAV relatively interacts less frequently with others in all three datasets. It is assigned the role of either servicing the southwest distant region in NCSU, or shuttling between remote areas in KAIST and Beijing. Therefore, frequency communication may not benefit the overall QoI index.

F. Computational Complexity Analysis

In practice, the model inference time is critical in metaverse applications. The time complexity of MetaCS can be theoretically calculated by:

TABLE V: Time cost in one-time model inference (ms).

Dataset	MetaCS	HATRPO	I2Q	t-LocPred	MAIC	GCRL-min(AoI)
NCSU	3.471	2.106	2.067	6.176	2.711	19.869
KAIST	3.632	2.274	2.078	6.911	2.738	38.510
Beijing	3.831	2.353	2.092	7.470	2.765	55.204

$$O\left(\sum_{i=1}^{N_{\text{MLP}}} d_{i,\text{in}} \cdot d_{i,\text{out}} + N_{\text{block}}(L^2 \cdot Z \cdot d + L \cdot Z^2 \cdot d) + U \cdot d^2\right), \tag{29}$$

where N_{MLP} denote the number of linear layers in MetaCS, $d_{i,\text{in}}$ and $d_{i,\text{out}}$ represent the dimension of input and output features of i-th linear layers; d donetes the embedding dimension in both **Pred** and **Colla**; The second term is for **Pred**, N_{block} denotes the number of encoders, L, Z, denote the length of $\mathbf{X}$ and the number of regions, respectively; The third term is for **Colla**, $O(U \cdot d^2)$ denotes the computational complexity of Bi-LSTM, where U is the number of UAVs.

Then, we demonstrate the magnitude of MetaCS's inference time by comparing with baselines. Table V shows the time cost from UAVs receive an observation to their actions are produced. When testing, we set $N_{\text{block}} = 3$, $L = 6$, $Z = 36$, $d = 64$, $U = 5$. We see that MetaCS is significantly faster than GCRL-min(AoI) and t-LocPred. This is because GCRL-min(AoI) employs Monte-Carlo tree search, resulting in exponential increase of computational complexity with more UAVs. t-LocPred uses LSTM to capture the temporal dependencies of user movements, unlike our attention modules in **Pred** which can be computed in parallel. Although MetaCS performs slightly slower than I2Q, HATRPO, and MAIC, it still maintains millisecond-scale speed, even when applied to large datasets, where the difference between other approaches are negligible in practice.

VII. Conclusion

In this paper, we proposed MetaCS, a user behavior prediction and UAV selective collaboration framework, for providing satisfactory QoI experiences of underlying metaverse users by collecting their data in MU-MIMO uplink channels. Specifically, we explicitly defined a quantitative overall QoI index, consists of both throughput and AoI satisfactory ratios. Based on IPPO, a Transformer-based spatial-temporal self-attention module is proposed to simultaneously incorporate geographic and semantic neighborhood information between regions. Also, a relational graph learning mechanism is introduced to enable the selection of the most informative partners for each UAV. Extensive results and trajectory visualization on real mobility datasets in NCSU, KAIST and Beijing demonstrate that MetaCS outperforms all six baselines in terms of overall QoI index.

References

[1] E. Bastug, M. Bennis, M. Medard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 110–117, 2017.

[2] F. Hu, Y. Deng, W. Saad, M. Bennis, and A. H. Aghvami, "Cellular-connected wireless virtual reality: Requirements, challenges, and solutions," *IEEE Communications Magazine*, vol. 58, no. 5, pp. 105–111, 2020.

[3] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A survey on mobile augmented reality with 5g mobile edge computing: Architectures, applications, and technical aspects," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 2, pp. 1160–1192, 2021.

[4] Microsoft, "Hololens 2—overview, features, and specs - microsoft," https://www.microsoft.com/en-us/hololens/hardware#document-experiences.

[5] C. H. Liu, P. Hui, J. W. Branch, C. Bisdikian, and B. Yang, "Efficient network management for context-aware participatory sensing," in *IEEE SECON'11*, 2011, pp. 116–124.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] W. Wang, Y. Yang, Z. Xiong, and D. Niyato, "Footstone of metaverse: a timely and secure crowdsensing," *IEEE Network*, 2023.

[11] Y. Han, D. Niyato, C. Leung, D. I. Kim, K. Zhu, S. Feng, X. Shen, and C. Miao, "A dynamic hierarchical framework for iot-assisted digital twin synchronization in the metaverse," *IEEE Internet of Things Journal*, vol. 10, no. 1, pp. 268–284, 2022.

[12] 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.

[13] C. H. Liu, Z. Dai, H. Yang, and J. Tang, "Multi-task-oriented vehicular crowdsensing: A deep learning approach," in *IEEE INFOCOM'20*, 2020, pp. 1123–1132.

[14] Y. Ye, C. H. Liu, Z. Dai, J. Zhao, Y. Yuan, G. Wang, and J. Tang, "Exploring both individuality and cooperation for air-ground spatial crowdsourcing by multi-agent deep reinforcement learning," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 205–217.

[15] F. Tang, X. Chen, M. Zhao, and N. Kato, "The roadmap of communication and networking in 6g for the metaverse," *IEEE Wireless Communications*, 2022.

[16] W. Xu, Y. Sun, R. Zou *et al.*, "Throughput maximization of uav networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 881–895, 2022.

[17] 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.

[18] 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.

[19] C. S. de Witt, T. Gupta, D. Makoviichuk *et al.*, "Is independent learning all you need in the starcraft multi-agent challenge?" *CoRR*, vol. abs/2011.09533, 2020.

[20] S. Ruiz-Correa, D. Santani, B. Ramirez-Salazar *et al.*, "Sensecityvity: Mobile crowdsourcing, urban awareness, and collective action in mexico," *IEEE Pervasive Computing*, vol. 16, no. 2, pp. 44–53, 2017.

[21] Z. Zhou, X. Li, C. You, K. Huang, and Y. Gong, "Joint sensing and communication-rate control for energy efficient mobile crowd sensing," *IEEE Transactions on Wireless Communications*, vol. 22, no. 2, pp. 1314–1327, 2023.

[22] F. Li, J. Zhao, D. Yu, X. Cheng, and W. Lv, "Harnessing context for budget-limited crowdsensing with massive uncertain workers," *IEEE/ACM Transactions on Networking*, vol. 30, no. 5, pp. 2231–2245, 2022.

[23] H. Wang, C. H. Liu, Z. Dai, J. Tang, and G. Wang, "Energy-efficient 3d vehicular crowdsourcing for disaster response by distributed deep reinforcement learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 3679–3687.

[24] D. Liu, Z. Du, X. Liu *et al.*, "Task-based network reconfiguration in distributed uav swarms: A bilateral matching approach," *IEEE/ACM Transactions on Networking*, pp. 1–13, 2022.

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

[26] 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.

[27] F. A. Oliehoek, C. Amato *et al.*, *A concise introduction to decentralized POMDPs*. Springer, 2016, vol. 1.

[28] Z. Chen, Y. Zhou, R.-R. Chen, and S. Zou, "Sample and communication-efficient decentralized actor-critic algorithms with finite-time analysis," in *ICML'22*, 2022, pp. 3794–3834.

[29] J. Xu, F. Zhong, and Y. Wang, "Learning multi-agent coordination for enhancing target coverage in directional sensor networks," *NeurIPS'20*, vol. 33, pp. 10 053–10 064, 2020.

[30] J. Jiang and Z. Lu, "The emergence of individuality," in *ICML'21*, 2021, pp. 4992–5001.

[31] C. Yu, A. Velu, E. Vinitsky *et al.*, "The surprising effectiveness of MAPPO in cooperative, multi-agent games," *CoRR*, vol. abs/2103.01955, 2021.

[32] X. Hao, H. Mao, W. Wang *et al.*, "Breaking the curse of dimensionality in multiagent state space: A unified agent permutation framework," in *ICLR'23*, 2023.

[33] Z. Dai, C. H. Liu, Y. Ye, R. Han *et al.*, "Aoi-minimal uav crowdsensing by model-based graph convolutional reinforcement learning," in *IEEE INFOCOM'22*, 2022, pp. 1029–1038.

[34] J. Zhang, Y. Zeng, and R. Zhang, "Multi-antenna uav data harvesting: Joint trajectory and communication optimization," *Journal of Communications and Information Networks*, vol. 5, no. 1, pp. 86–99, 2020.

[35] J. Li, H. Zhao, H. Wang *et al.*, "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.

[36] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser mimo systems," *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1436–1449, 2013.

[37] H. Huh, A. M. Tulino, and G. Caire, "Network mimo with linear zero-forcing beamforming: Large system analysis, impact of channel estimation, and reduced-complexity scheduling," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 2911–2934, 2011.

[38] T. K. Lo, "Maximum ratio transmission," in *IEEE ICC'99*, vol. 2, 1999, pp. 1310–1314.

[39] Y. Zeng, X. Xu, and R. Zhang, "Trajectory design for completion time minimization in uav-enabled multicasting," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2233–2246, 2018.

[40] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *IEEE SECON'11*, 2011, pp. 350–358.

[41] B. Krogfoss, J. Duran, P. Perez, and J. Bouwen, "Quantifying the value of 5g and edge cloud on qoe for ar/vr," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2020, pp. 1–4.

[42] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," in *NeurIPS'17*, 2017, p. 6000–6010.

[43] M. Müller, "Dynamic time warping," *Information Retrieval for Music and Motion*, pp. 69–84, 2007.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR'16*, 2016, pp. 770–778.

[45] A. Singh, T. Jain, and S. Sukhbaatar, "Learning when to communicate at scale in multiagent cooperative and competitive tasks," *arXiv preprint arXiv:1812.09755*, 2018.

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

[47] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *13th International Conference on Artificial Intelligence and Statistics (AISTATS'10)*, 2010, pp. 249–256.

[48] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[49] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 316–324.

[50] J. Jiang and Z. Lu, “I2q: A fully decentralized q-learning algorithm,” *NeurIPS’ 22*, 2022.
[51] J. G. Kuba, R. Chen, M. Wen *et al.*, “Trust region policy optimisation in multi-agent reinforcement learning,” in *ICLR’22*, 2022.
[52] L. Yuan, J. Wang, F. Zhang, C. Wang, Z. Zhang, Y. Yu, and C. Zhang, “Multi-agent incentive communication via decentralized teammate modeling,” in *AAAI’22*, 2022, pp. 9466–9474.
[53] C. H. Liu, Y. Wang, C. Piao, Z. Dai *et al.*, “Time-aware location prediction by convolutional area-of-interest modeling and memory-augmented attentive lstm,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 2472–2484, 2022.
[54] R. Jiang, Z. Cai, Z. Wang, C. Yang, Z. Fan, Q. Chen, K. Tsubouchi, X. Song, and R. Shibasaki, “Deepcrowd: A deep model for large-scale citywide crowd density and flow prediction,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 276–290, 2021.

Yuxiao Ye receives a BSc degree in Computer Science from Beijing Institute of Technology, China, in 2022. He is currently working toward an MSc degree under the supervision of Prof. Chi Harold Liu at the School of Computer Science and Technology, Beijing Institute of Technology, China. He is now working on the problems of mobile crowdsensing and deep reinforcement learning.

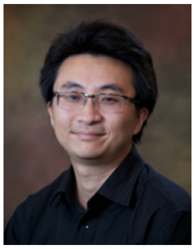
Hao Wang receives a B.Eng degree in Software Engineering from Beijing Institute of Technology, China, in 2021. He is currently working toward an MSc degree under the supervision of Prof. Chi Harold Liu at the School of Computer Science and Technology at Beijing Institute of Technology, China. He has published a few works at IEEE ToN, INFOCOM and ACM SIGKDD, and is now working on the problems of mobile crowdsensing and deep reinforcement learning.

Chi Harold Liu (SM’15) receives a Ph.D. degree in Electronic Engineering from Imperial College, UK in 2010, and a B.Eng. degree in Electronic and Information Engineering from Tsinghua University, China in 2006. He is currently a Full Professor and Vice Dean at the School of Computer Science and Technology, Beijing Institute of Technology, China. He has worked for IBM Research - China and Deutsche Telekom Laboratories. His current research interests include mobile crowdsensing by deep learning. He received the IBM First Plateau Invention Achievement Award in 2012, ACM SigKDD’21 Best Paper Runner-up Award, and ACM MobiCom’21 Best Community Paper Runner-up Award. He serves as the Associate Editor for IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING. He is a senior member of IEEE and a Fellow of IET and British Computer Society.

Zipeng Dai is currently a PhD student under the supervision of Prof. Chi Harold Liu in the School of Computer Science and Technology at Beijing Institute of Technology, China. He has published a few works at IEEE INFOCOM, TMC and ACM SIGKDD, and is now working on the problem of mobile crowdsensing with deep reinforcement learning.

Guozheng Li received his Ph.D. degree in Computer Science from the School of EECS, Peking University, in 2021. He is an assistant professor at the School of Computer Science and Technology, Beijing Institute of Technology, China. His primary research interests include information visualization, especially hierarchical data visualization, and visualization authoring.

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



Jian Tang (F’19) received the PhD degree in computer science from Arizona State University, in 2006. He is with Midea Group, China and a Full Professor in the Department of Electrical Engineering and Computer Science, Syracuse University. His research interests lie in the areas of cloud computing, big data and machine learning. He has published more than 100 papers in premier journals and conferences. He received an NSF CAREER award in 2009, the 2016 Best Vehicular Electronics Paper Award from IEEE Vehicular Technology Society, and Best Paper Awards from IEEE ICC’14 and IEEE Globecom’15 respectively. He has been an editor for the IEEE Transactions on Wireless Communications since 2016, for the IEEE Transactions on Vehicular Technology since 2010, for IEEE Transactions on Mobile Computing since 2017, and for the IEEE Internet of Things Journal since 2013. He served as a TPC co-chair for IEEE iThings’15 and IEEE ICNC’16. He also served as a TPC member for many international conferences, including IEEE INFOCOM 2010-2018, ICDCS 2015, ICC 2006-2016, Globecom 2006-2016, etc. He is a Fellow of IEEE.