Quantum DRL for Green UAV Positioning in 6G-Enabled SAGIN with Cooperative Nano-Satellite Constellations

Sasinda C. Prabhashana*, Dang Van Huynh *, Octavia A. Dobre *, Hyundong Shin†, and Trung Q. Duong*‡
* Memorial University, Canada, e-mails: {cwelhengodag,vdhuynh, odobre, tduong}@mun.ca
† Kyung Hee University, South Korea, e-mail: hshin@khu.ac.kr
‡ Queen's University Belfast, UK, e-mail: trung.q.duong@qub.ac.uk

***Abstract*—In this paper, we explore a 6G-enabled space-air-ground integrated network (SAGIN) framework that integrates ground communication hubs (CHs), a UAV with mobile edge computing (MEC) capabilities, and a constellation of low Earth orbit (LEO) nano-satellites. We formulate a joint optimization problem for UAV trajectory, task offloading, and satellite load balancing, modeled as a mixed-integer nonlinear programming (MINLP) problem. To solve this, we propose a quantum-enhanced advantage actor-critic (QEA2C) reinforcement learning algorithm that employs quantum neural networks and two quantum state encoding methods: amplitude encoding (AE) and higher-order encoding (HOE). Simulation results show that HOE achieves superior performance in terms of convergence speed, cumulative rewards, and learning efficiency, successfully serving all CHs with a well-optimized UAV trajectory. Meanwhile, AE achieves better cost minimization with lower resource consumption, making it a more practical option when computational efficiency is a priority. Moreover, these results highlight the trade-offs between learning performance and cost efficiency in quantum-enhanced decision-making for managing 6G-enabled SAGINs.**

I. Introduction

Global connectivity is a key objective of future sixth-generation (6G) communication systems. A primary challenge lies in achieving reliable, seamless coverage across terrestrial, aerial, and space domains. The space-air-ground integrated network (SAGIN) has emerged as a promising architecture to connect ground infrastructure, aerial platforms, and satellite networks [1]. Recently, satellite networks have become vital for providing reliable global access services in SAGIN [2], [3]. However, large-scale satellite deployment remains challenging due to the high cost and complexity of launch and maintenance. As a practical alternative, CubeSats have emerged as a cost-effective solution [1]. These satellites, operating in low Earth orbit, offer significantly lower propagation delays than GEO satellites. In particular, nano-satellites, a subclass of CubeSats, have attracted attention for their lightweight design, affordability, and ease of deployment. Their scalability and support for ubiquitous connectivity make them well suited for modern space-based networks [1], [3].

Furthermore, unmanned aerial vehicles (UAVs) are a key component of SAGINs, offering flexible coverage, particularly in areas lacking ground infrastructure [2]. However, their limited range and energy constraints necessitate efficient trajectory planning to ensure reliable connectivity and minimize latency [4]. Integrating UAVs with mobile edge computing (MEC) enables real-time data processing closer to end users, reducing latency. When combined with ultra-reliable low-latency communication (URLLC), UAV networks become capable of supporting mission critical applications with stringent delay and reliability requirements [5].

Integrating these components into a single network while maintaining reliability, high data rates, and efficient spectrum access remains a critical challenge. Quantum computing presents promising solutions by exploiting superposition and parallelism to rapidly process large and complex state spaces, making it well suited for dynamic communication environments [6]–[8]. Moreover, integrating quantum computing with deep reinforcement learning (DRL) enhances decision making efficiency [9]. However, scaling quantum-enhanced DRL for practical communication scenarios remains challenging. Therefore, further research is essential to advance quantum DRL and fully realize its advantages in dynamic communication environments.

In this paper, we propose a 6G-enabled SAGIN framework integrating communication hubs, a UAV with MEC capabilities, and LEO nano satellites. We formulate a cost minimization problem jointly optimizing UAV trajectory, task offloading, and satellite load distribution modeled as a mixed-integer nonlinear programming (MINLP) and solved using a quantum-enhanced advantage actor critic (QEA2C) algorithm. We evaluate amplitude encoding (AE) and higher-order encoding (HOE) under the proposed QEA2C framework. Simulations show that HOE leads to faster convergence, higher rewards, and successfully serves all CHs, while AE achieves lower overall system cost with better energy and cost efficiency.

II. System Model

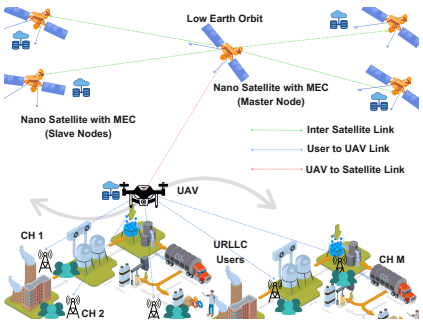


Fig. 1: An illustration of 6G-Enabled space-air-ground integrated network with nano-satellite constellation

In this paper, we consider a 6G-enabled SAGIN with nano-satellite constellations, as illustrated in Fig. 1. The ground layer consists of M CHs, denoted by $\mathcal{M} = \{1, 2, \ldots, M\}$. Each

CH collects sensor data, forms data packets, and transmits them under URLLC constraints to the UAV, where a UAV with MEC capabilities performs computation. The UAV uses K number of antennas to communicate with CHs and it flies over the ground layer to collect the task from each CH. However, due to multifunctional operations, the UAV experiences substantial energy consumption and faces significant challenges in managing task-request overflow during peak times. To address this, the UAV offloads excess tasks to a LEO nano-satellite network which is in the space layer. This network is configured as in a hub-and-spoke constellation and comprises a master nano-satellite (MNS) and four slave nano-satellites (SNS), which are denoted by $\mathcal{I} = \{1, 2, 3, 4\}$. Each nano-satellite is equipped with a single antenna and MEC capabilities. The MNS coordinates and manages computational load distribution among the SNSs. Upon receiving computation intensive and latency sensitive data from CHs, the UAV transmits a portion of this data to the MNS, which then distributes it to SNSs for processing with load balancing. This network is modeled using a three-dimensional (3D) Cartesian coordinate system. The CHs are located on the ground at $\ell_{\mathrm{m}} = (x_{\mathrm{m}}, y_{\mathrm{m}}, 0) \in \mathbb{R}^3$, for $\mathrm{m} \in \mathcal{M}$, while the UAV operates at a fixed altitude H with position $\ell_{\mathrm{u}}(t) = (x_{\mathrm{u}}(t), y_{\mathrm{u}}(t), H) \in \mathbb{R}^3$. In the space layer, the MNS is positioned at $\ell_{\mathrm{ms}}(t) = (x_{\mathrm{ms}}(t), y_{\mathrm{ms}}(t), R) \in \mathbb{R}^3$, where R denotes the LEO orbit altitude. The four SNSs are arranged at the corners of a square centered around the MNS, each at an equal distance. Moreover, we assume continuous satellite coverage due to the dense deployment of LEO satellites.

A. Channel Modeling

1) Channel between CH and UAV: The channel vector between the m-th CH and UAV at time t can be denoted as [5]

$$\mathbf{g}_{\mathrm{m,u}}(t) = \sqrt{h_{\mathrm{m,u}}(t)}\tilde{\mathbf{g}}_{\mathrm{m,u}}(t) \in \mathbb{C}^{K\times 1}, \tag{1}$$

where $\tilde{\mathbf{g}}_{\mathrm{m,u}}(t)$ is the small-scale fading which follows the complex random variable with $\mathbb{E}\left[|\tilde{g}_{\mathrm{m,u}}(t)|^2\right] = 1$. Furthermore, $h_{\mathrm{m,u}}(t)$ is the large-scale channel coefficient, which can be calculated as [3]

$$h_{\mathrm{m,u}}(t) = \left(\frac{4\pi f_c d_{\mathrm{m,u}}(t)}{c}\right)^{-\alpha} 10^{-\frac{\kappa^{\mathrm{los}}_{\mathrm{m,u}}(t)\eta^{\mathrm{los}} + \kappa^{\mathrm{nlos}}_{\mathrm{m,u}}(t)\eta^{\mathrm{nlos}}}{10}}, \tag{2}$$

where f_c is the carrier frequency and c denotes the speed of light. Moreover, α is the path loss exponent and $d_{\mathrm{m,u}}(t)$ is the distance between m-th CH and the UAV at time t. It can be calculated as $d_{\mathrm{m,u}}(t) = \left((x_{\mathrm{u}}(t) - x_{\mathrm{m}})^2 + (y_{\mathrm{u}}(t) - y_{\mathrm{m}})^2 + H^2\right)^{1/2}$. Furthermore, $\kappa^{\mathrm{los}}_{\mathrm{m,u}}(t)$ is the probability of LoS component which can be calculated as

$$\kappa^{\mathrm{los}}_{\mathrm{m,u}}(t) = \left(1 + \mu_1 \exp\left[-\mu_2\left(\arctan\left(H/d^{\mathrm{hd}(t)}_{\mathrm{m,u}}\right) - \mu_1\right)\right]\right)^{-1}, \tag{3}$$

where μ_1 and μ_2 coefficients depend on the environment. Moreover, probability of the NLoS component can be calculated as $\kappa^{\mathrm{nlos}}_{\mathrm{m,u}}(t) = 1 - \kappa^{\mathrm{los}}_{\mathrm{m,u}}(t)$. Furthermore, η^{los} and η^{nlos} represent the excessive path loss for LoS and NLoS components respectively. $d^{\mathrm{hd}}_{\mathrm{m,u}}(t)$ is the horizontal distance between the m-th CH and the UAV which can be calculated as $d^{\mathrm{hd}}_{\mathrm{m,u}}(t) = \left((x_{\mathrm{u}}(t) - x_{\mathrm{m}})^2 + (y_{\mathrm{u}}(t) - y_{\mathrm{m}})^2\right)^{1/2}$.

2) Channel between UAV to MNS: The channel vector between the UAV and MNS at time t can be denoted as [3]

$$\mathbf{g}_{\mathrm{u,ms}}(t) = \begin{bmatrix} \mathrm{g}_1(t) & \mathrm{g}_2(t) & \dots & \mathrm{g}_K(t) \end{bmatrix} \in \mathbb{C}^{1\times K}. \tag{4}$$

Here, each component $\mathrm{g}_{\mathrm{x}}(t)$ is characterized by the shadowed-Rician fading model which can be expressed as $\mathrm{g}_{\mathrm{x}}(t) = \sqrt{h_{\mathrm{x}}(t)}d^{-\alpha/2}_{\mathrm{u,ms}}(t)$, where $d_{\mathrm{u,ms}}(t)$ is the distance between the UAV and the MNS, which can be calculated as $d_{\mathrm{u,ms}}(t) = \left((x_{\mathrm{ms}}(t) - x_{\mathrm{u}}(t))^2 + (y_{\mathrm{ms}}(t) - y_{\mathrm{u}}(t))^2 + (R - H)^2\right)^{1/2}$. Furthermore, $h_{\mathrm{x}}(t)$ follows a shadowed-Rician distribution, which can be denoted as $\mathrm{g}^{\mathrm{x}}_{\mathrm{u,ms}}(t) \sim \mathrm{SR}(\Omega_{\mathrm{x}}, \Delta_{\mathrm{x}}, \epsilon_{\mathrm{x}})$, where Ω_{x} is the average power of the direct signal, Δ_{x} is the half-average power of the scattered multipath component, and ϵ_{x} represents the Nakagami-m fading parameter.

3) Channel Between MNS to SNS: The channel between the MNS and the i-th SNS at time t can be expressed as [3]

$$\mathbf{g}_{\mathrm{ms,ss(i)}}(t) = \sqrt{h_{\mathrm{ms,ss(i)}}(t)}\tilde{g}_{\mathrm{ms,ss(i)}}(t) \in \mathbb{C}^{1\times 1}, \tag{5}$$

where $h_{\mathrm{ms,ss(i)}}(t)$ is the large-scale path loss, which can be calculated as $h_{\mathrm{ms,ss(i)}}(t) = 10^{-\frac{\kappa_{\mathrm{ms,ss(i)}}(t)}{10}}$. Here, $\kappa_{\mathrm{ms,ss(i)}}(t)$ is the free-space path loss, which can be calculated as $\kappa_{\mathrm{ms,ss(i)}}(t) = 10\log\left(\frac{4\pi f_c d_{\mathrm{ms,ss(i)}}(t)}{c}\right)^{\alpha}$, where $d_{\mathrm{ms,ss(i)}}(t)$ is the distance between the MNS and the i-th SNS, which can be calculated as $d_{\mathrm{ms,ss(i)}}(t) = \left((x_{\mathrm{ms}}(t) - x_{\mathrm{ss(i)}}(t))^2 + (y_{\mathrm{ms}}(t) - y_{\mathrm{ss(i)}}(t))^2\right)^{1/2}$. Moreover, $\tilde{g}_{\mathrm{ms,ss(i)}}(t)$ is the small-scale fading which can be characterized as $\sim \mathcal{CN}(0, 1)$.

B. Communication Modeling

1) Communication between CH and UAV: The data from each CH is received by the UAV. Therefore, the data rate between the m-th CH and UAV under URLLC constraints can be calculated as

$$\mathcal{R}^{\mathrm{u}}_{\mathrm{m}}(t) \approx \mathcal{B}\left(\log_2\left(1 + \gamma^{\mathrm{u}}_{\mathrm{m}}(t)\right) - \sqrt{\frac{V^{\mathrm{u}}_{\mathrm{m}}(t)}{N}}\frac{Q^{-1}(\epsilon^{\mathrm{u}}_{\mathrm{m}})}{\ln 2}\right), \tag{6}$$

where $\mathcal{B}$ is the system bandwidth. N is the length of the block. The parameter $\epsilon^{\mathrm{u}}_{\mathrm{m}}$ corresponds to the probability of decoding errors. The function $Q^{-1}(\cdot)$ refers to the inverse of the function Q, which is defined as $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty}\exp\left(-\frac{t^2}{2}\right)dt$. $V^{\mathrm{u}}_{\mathrm{m}}(t)$ represents the channel dispersion, which is given by $V^{\mathrm{u}}_{\mathrm{m}}(t) = 1 - \left[1 + \gamma^{\mathrm{u}}_{\mathrm{m}}(t)\right]^{-2}$, where $\gamma^{\mathrm{u}}_{\mathrm{m}}(t)$ is the signal-to-noise ratio where it can be calculated as $\gamma^{\mathrm{u}}_{\mathrm{m}}(t) = p_{\mathrm{m}}|\mathbf{g}_{\mathrm{m,u}}(t)|^2/\sigma^2_{\mathrm{u}}(t)$, where p_{m} represents the transmission power of the m-th CH and $\sigma_{\mathrm{u}}(t)$ is the instantaneous noise power, characterized by a Gaussian complex normal distribution $\sim \mathcal{CN}\left(0, \sigma^2\right)$ [5].

2) Communication between UAV and MNS: The data rate from the UAV to MNS can be calculated as

$$\mathcal{R}^{\mathrm{ms}}_{\mathrm{u}}(t) = \mathcal{B}\log_2\left(1 + \frac{p_{\mathrm{u}}|\mathbf{g}_{\mathrm{u,ms}(t)}|^2}{\sigma^2_{\mathrm{ms}}(t)}\right), \tag{7}$$

where $\sigma_{\mathrm{ms}}(t)$ is the instantaneous noise power, followed by Gaussian complex normal distribution $\sim \mathcal{CN}\left(0, \sigma^2\right)$ [4]. Moreover, p_{u} is the transmission power of the UAV.

3) Communication between MNS and SNS: The data rate from the MNS to i-th SNS can be expressed as

$$\mathcal{R}_{\text{ms}}^{\text{ss(i)}}(t) = \mathcal{B}\log_2\left(1 + \frac{p_{\text{ms,ss(i)}}|\mathbf{g}_{\text{ms,ss(i)}}(t)|^2}{\sigma^2_{\text{ss(i)}(t)}}\right), \tag{8}$$

where $p_{\text{ms,ss(i)}}$ is the transmission power of the MNS to i-th SNS. $\sigma_{\text{ss(i)}}(t)$ is the instantaneous noise power, characterized by Gaussian complex normal distribution $\sim \mathcal{CN}\left(0, \sigma^2\right)$ [4].

C. Mobility of UAV

We consider that the total flying time of the UAV into T discrete time slots each of length τ_t, which can be denoted as $\mathcal{T} = \{1, 2, \cdots, T\}$. Moreover, the UAV moves with a fixed altitude H. At time t, the UAV's movement is determined by the displacement components $d_x(t)$ and $d_y(t)$ along the x- and y-axes, respectively. Therefore, UAV's position from time slot t to $t+1$, can be calculated as [4]

$$\begin{aligned} x_{\text{u}}(t+1) &= x_{\text{u}}(t) + d_x(t) + \Delta x_{\text{u}}, \\ y_{\text{u}}(t+1) &= y_{\text{u}}(t) + d_y(t) + \Delta y_{\text{u}}, \end{aligned} \tag{9}$$

where Δx_{u} and Δy_{u} are small random disturbances due to wind, modeled as $\sim \mathcal{N}(0, \sigma^2)$. The instantaneous velocity $v(t)$ of the UAV at time t is given by $v(t) = d_{\text{xy}}(t)/\tau_t$, where $d_{\text{xy}}(t) = \sqrt{d_x(t)^2 + d_y(t)^2}$. Consequently, the UAV's power consumption can be expressed as

$$\begin{aligned} P(v_{\text{u}}(t)) = \chi_1\left(1 + \frac{3(v_{\text{u}}(t))^2}{w_{\text{tip}}^2}\right) + \chi_2\left(\sqrt{1 + \frac{(v_{\text{u}}(t))^4}{4w_0^4}} - \frac{(v_{\text{u}}(t))^2}{2w_0^2}\right)^{\frac{1}{2}} \\ + \frac{1}{2}\tau\rho\Upsilon A(v_{\text{u}}(t))^3, \end{aligned} \tag{10}$$

where A is the area of the rotor disc, Υ denotes the rotor solidity, ρ denotes the air density, and τ is the fuselage drag coefficient. Moreover, w_0 represents the average rotor-induced flow in hover, while w_{tip} denotes the tip speed of the rotor blades. The χ_1 and χ_2 coefficients for the blade profile power and the induced power required in hover, respectively. Therefore, total energy consumption during the time T can be calculated as $E_{\text{u}}^{\text{fly}}(t) = \sum_{t=1}^{T} P(v_{\text{u}}(t))\tau_t$ [9].

D. Energy Aware Task Scheduling, Transmission and Processing Model

We consider that each CH generate a task at time t, denoted as $J_{\text{m}} = \{D_{\text{m}}, Q_{\text{m}}, T_{\text{m}}^{\max}\}$, where D_{m} is the data size, Q_{m} the computational complexity, and $T_{\text{m}}^{\max}$ the maximum latency. we define a binary variable $\pi_{\text{m}}(t) \in \{0, 1\}$ to indicates task acceptance by the UAV. If the CH is within range $d_{\text{m,u}}^{\text{h}}(t) < d^{\max}$ the CH transmits the data to the UAV. Therefore, energy consumption for transmission can be calculated as $E_{\text{m,u}}^{\text{trans}}(t) = \pi_{\text{m}}(t) p_{\text{m}} D_{\text{m}}/\mathcal{R}_{\text{m}}^{\text{u}}(t)$, where p_{m} is the CH's transmission power. Moreover, UAV processes fraction of the task locally and offloads the rest to the MNS. Let $\delta_{\text{m}}(t) \in [0, 1]$ represent the offloaded fraction. Then, UAV's processing energy can be calculated as $E_{\text{m,u}}^{\text{proc}}(t) = \pi_{\text{m}}(t)\xi_{\text{u}}(1 - \delta_{\text{m}}(t))Q_{\text{m}}(f^{\text{u}})^2$, where ξ_{u} is the energy coefficient and f^{u} the allocated frequency. The offloading energy to MNS can be expressed as $E_{\text{u,ms}}^{\text{trans}}(t) = \pi_{\text{m}}(t) p_{\text{u}} \delta_{\text{m}}(t) D_{\text{m}}/\mathcal{R}_{\text{u}}^{\text{ms}}(t)$. Concurrently, the MNS distributes the received fraction to SNSs using load-balancing variables $\theta_{\text{m,i}}(t) \in [0, 1]$, with transmission energy to i-th SNS can be calculated as $E_{\text{ms,ss}(i)}^{\text{trans}}(t) = \pi_{\text{m}}(t)\theta_{m,i}(t)\, p_{\text{ms,ss}(i)}\delta_{\text{m}}(t)\, D_{\text{m}}/\mathcal{R}_{\text{ms}}^{\text{ss}(i)}(t)$ and processing energy as $E_{\text{ss}(i)}^{\text{proc}}(t) = \pi_{\text{m}}(t)\theta_{\text{m,i}}(t)\xi_{\text{ss}(i)} Q_{\text{m}}\delta_{\text{m}}(t)(f^{\text{ss}(i)})^2$, where $\xi_{\text{ss}(i)}$ and $f^{\text{ss}(i)}$ are the energy coefficient and computational frequency of i-th SNS. The total space layer energy consumption can be calculated as $E_{\text{ms,ss}}^{\text{total}}(t) = \sum_{i=1}^{4}\left(E_{\text{ms,ss}(i)}^{\text{trans}}(t) + E_{\text{ss}(i)}^{\text{proc}}(t)\right)$, and the total energy for processing the m-th CH task across all SAGIN layers can be calculated as

$$E_{\text{m}}^{\text{total}}(t) = E_{\text{m,u}}^{\text{trans}}(t) + E_{\text{m,u}}^{\text{proc}}(t) + E_{\text{u,ms}}^{\text{trans}}(t) + E_{\text{ms,ss}}^{\text{total}}(t). \tag{11}$$

E. Latency Model

When the UAV accepts a task from the m-th CH, the transmission latency from the m-th CH to UAV can be expressed as $T_{\text{m,u}}^{\text{trans}}(t) = \pi_{\text{m}}(t) D_{\text{m}}/\mathcal{R}_{\text{m}}^{\text{u}}(t)$. Furthermore, the processing latency at the UAV for the locally processed portion of m-th CH task can be calculated as $T_{\text{m,u}}^{\text{proc}}(t) = \pi_{\text{m}}(t)(1-\delta_{\text{m}}(t))Q_{\text{m}}/f^{\text{u}}$. Furthermore, the transmission latency for the offloaded portion from the UAV to the MNS can be expressed as $T_{\text{u,ms}}^{\text{trans}}(t) = \pi_{\text{m}}(t)\delta_{\text{m}}(t) D_{\text{m}}/\mathcal{R}_{\text{u}}^{\text{ms}}(t)$. Once the MNS receives the offloaded task from UAV, it distributes the task portion among the all the SNSs. Therefore, the transmission latency from the MNS to the i-th SNS can be calculated as $T_{\text{ms,ss}(i)}^{\text{trans}}(t) = \pi_{\text{m}}(t)\theta_{\text{m,i}}(t)\delta_{\text{m}}(t) D_{\text{m}}/\mathcal{R}_{\text{ms}}^{\text{ss}(i)}(t)$ and the corresponding processing latency at the i-th SNS can be expressed as $T_{\text{ss}(i)}^{\text{proc}}(t) = \pi_{\text{m}}(t)\theta_{\text{m,i}}(t)\delta_{\text{m}}(t) Q_{\text{m}}/f^{\text{ss}(i)}$. Consequently, the total latency for the task of the m-th CH can be calculated as

$$\begin{aligned} T_{\text{m}}^{\text{total}}(t) = T_{\text{m,u}}^{\text{trans}}(t) + T_{\text{m,u}}^{\text{proc}}(t) + T_{\text{u,ms}}^{\text{trans}}(t) + \\ \max_{i\in\{1,2,3,4\}}\left(T_{\text{ms,ss}(i)}^{\text{trans}}(t) + T_{\text{ss}(i)}^{\text{proc}}(t)\right) \end{aligned} \tag{12}$$

F. Problem Formulation

In this paper, we aim to jointly optimize the UAV's trajectory, task offloading decisions, and satellite load distribution to minimize the total system cost across the SAGIN layers. The cost includes the UAV's flying energy, task-related energy consumption, and latency over T time slots. The optimization ensures that all tasks from CHs are completed within the UAV's battery capacity $E_{\text{u}}^{\text{batt}}(t)$. Accordingly, the objective function can be expressed as follows:

$$f(\Omega) = \sum_{t=1}^{T} w_1 E_{\text{u}}^{\text{fly}}(t) + \sum_{t=1}^{T}\sum_{m=1}^{M}\left(w_2 E_{\text{m}}^{\text{total}}(t) + w_3 T_{\text{m}}^{\text{total}}(t)\right), \tag{13}$$

where $\Omega \triangleq \{\boldsymbol{\pi}, \boldsymbol{\delta}, \boldsymbol{\theta}, \mathbf{d}\}$ is the set of optimization variables, with $\boldsymbol{\pi} \triangleq \{\pi_{\text{m}}(t)\}_{\forall m}$, $\boldsymbol{\delta} \triangleq \{\delta_{\text{m}}(t)\}_{\forall m}$, $\boldsymbol{\theta} \triangleq \{\theta_{\text{m,i}}(t)\}_{\forall m,i}$, and $\mathbf{d} \triangleq \{d_x(t), d_y(t)\}_{\forall t}$. w_1, w_2, w_3 are the weighting factors. Therefore, the optimization problem can be formulated as follows:

$$\textbf{\textit{(P1):}} \quad \min_{\{\boldsymbol{\pi},\boldsymbol{\delta},\boldsymbol{\theta},\mathbf{d}\}} \quad f(\Omega) \tag{14a}$$

$$\text{s.t.} \quad 0 \le \delta_{\text{m}}(t) \le 1, \quad \forall m, t, \tag{14b}$$

$$\pi_{\mathrm{m}}(t) \in \{0,1\}, \quad \forall m,t, \tag{14c}$$

$$T_{\mathrm{m}}^{\mathrm{total}}(t) \le T_{\mathrm{m}}^{\max}, \quad \text{when } \pi_{\mathrm{m}}(t)=1, \quad \forall m,t, \tag{14d}$$

$$0 \le d_{\mathrm{xy}}(t) \le v^{\max}\tau_t, \quad \forall t, \tag{14e}$$

$$0 \le x_{\mathrm{u}}(t) \le x^{\max}, \quad 0 \le y_{\mathrm{u}}(t) \le y^{\max}, \quad \forall t, \tag{14f}$$

$$\pi_{\mathrm{m}}(t) d_{\mathrm{m,u}}^{\mathrm{hd}}(t) \le d^{\max}, \quad \forall m,t, \tag{14g}$$

$$\sum_{i=1}^{4} \theta_{\mathrm{m,i}}(t) = 1, \quad \forall m,t, \quad i \in \mathcal{I} \tag{14h}$$

$$\sum_{t=1}^{T} \pi_{\mathrm{m}}(t) = 1, \quad \forall m, \quad \sum_{m=1}^{M} \pi_{\mathrm{m}}(t) \le 1, \quad \forall t, \tag{14i}$$

$$0 \le \theta_{\mathrm{m,i}}(t) \le 1, \quad i \in \mathcal{I}, \quad \forall m,t, \tag{14j}$$

$$\mathcal{R}_{\mathrm{m}}^{\mathrm{u}}(t), \mathcal{R}_{\mathrm{u}}^{\mathrm{ms}}(t), \mathcal{R}_{\mathrm{ms}}^{\mathrm{ss(i)}}(t) \ge \mathcal{R}^{\min}, \quad \forall m,t, \tag{14k}$$

$$\sum_{t=1}^{T} E_u^{\mathrm{fly}}(t) \le E_{\mathrm{u}}^{\mathrm{batt}}. \tag{14l}$$

As specified in (14), the objective function (14a) minimizes the total cost, including flying energy, task-related energy, and latency. Constraints (14b)–(14d) regulate offloading fractions, binary task acceptance, and latency limits. Constraint (14e) restricts UAV movement per slot, while (14f) keeps it within the operational area. Constraints (14g)–(14j) ensure UAV–CH coverage, complete task distribution, single CH acceptance per slot, and valid SNS allocation. Constraints (14k) and (14l) enforce minimum data rates and limit flying energy within battery capacity.

III. Proposed Solution

A. Transformation to Reinforcement Learning Framework

The optimization problem in (14) is a mixed integer non-linear programming problem, which is challenging to solve using conventional methods. To overcome this, we propose a quantum-based DRL algorithm, namely quantum enhanced advantage actor critic (QEA2C), which leverages quantum parallelism to enhance exploration and convergence. To facilitate the DRL-based solution, the problem is formulated as a Markov Decision Process (MDP) which comprise with state space ($\mathcal{S}$), action space ($\mathcal{A}$) and Reward function ($\mathcal{R}$).

1) State space: The state space $\mathcal{S}$ comprises the system's status at each time slot t. It provides the QEA2C agent with sufficient information to make decisions. Therefore, the state space $s(t)$ at time t can be defined as

$$s(t) = \big[\, \{x_u(t),\, y_u(t)\},\, E_u^{\mathrm{remain}}(t),\, CH_{\mathrm{served}}(t),\, \{\Delta x_{\mathrm{near}}(t),\, \Delta y_{\mathrm{near}}(t)\},\, d_{\mathrm{near}}(t),\, I_{\mathrm{unserved}}(t) \,\big],$$

where $\{x_u(t),\, y_u(t)\}$ are the UAV's coordinates at time t, $E_u^{\mathrm{remain}}(t)$ is the UAV's remaining battery energy. Moreover, $CH_{\mathrm{served}}(t) = \frac{\sum_{m=1}^{M} \pi_m(t)}{M}$ is the percentage of CHs served at time t, $\{\Delta x_{\mathrm{near}}(t),\, \Delta y_{\mathrm{near}}(t)\}$ are the relative coordinates to the nearest unserved CH, $d_{\mathrm{near}}(t)$ is the distance to the nearest unserved CH, and $I_{\mathrm{unserved}}(t) \in \{0,1\}$ denotes the presence of unserved CHs at time t.

2) Action space: The action space $\mathcal{A}$ defines the agent's control decisions at each time slot t. Therefore at time t, the action space $a(t)$ can be denoted as

$$a(t) = [\pi(t), \delta(t), \theta(t), d_x(t), d_y(t)],$$

where $\pi(t)$ selects the CH index to serve. Moreover, $\delta(t)$ is the fraction of the selected CH's task offloaded to the MNS. Furthermore, $\theta_i(t)$, with (14h), distribute the tasks across all SNSs. $d_x(t)$, $d_y(t)$ are the UAV's displacement variables in x and y directions respectively.

3) Reward: The reward function $\mathcal{R}$ plays a key role in guiding the learning process of the QEA2E agent. It provides feedback that evaluates the quality of the agent's actions. Therefore, the reward function $r(t)$ can be expressed as

$$r(t) = \frac{1}{W}(-f(\Omega)) + B(t) - P(t)$$

where, W is the scaling factor of the reward function. Moreover, $B(t)$ denotes the bonus function which can defined as $B(t) = B(\text{served}) + B(\text{all served})$, where $B(\text{served})$ is reward function if a CH is served at time t, and $B(\text{all served})$ is reward function if all CHs are served; otherwise, $B(t) = 0$. Furthermore, $P(t)$ is the penalty functions designed to enforce the constraints (14a)–(14l).

B. Quantum Enhanced Advantage Actor Critic Algorithm

We employ the QEA2C algorithm, a quantum-enhanced, model free actor-critic DRL method, where classical states $s(t)$ are first encoded into quantum states $s_q(t)$ and then processed by quantum neural networks. In order to evaluate the performance of the quantum encoding techniques, we employ two encoding schemes namely AE and HOE and compare their effectiveness in the QEA2C framework. In AE, classical information is encoded in the amplitudes of a quantum state, allowing an n-dimensional vector $s(t) = [x_1, x_2, \ldots, x_n]$ to be represented using $\log_2 n$ qubits by leveraging quantum superposition. In the SAGIN environment, the classical state $s(t)$ is normalized as $x_{\mathrm{norm}} = x/\|x\|_2$, where $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$, ensuring $\langle s_q(t)|s_q(t)\rangle = 1$. The encoded quantum state is given by $|s_q(t)\rangle = \mathbf{U}_{\mathrm{AE}}^{x}|0\rangle^{\otimes m} \sum_{i=0}^{n-1} x_{\mathrm{norm},i}|i\rangle$, where $m = \log_2 n$, $|i\rangle$ is the i-th computational basis state, and $\mathbf{U}_{\mathrm{ae}}^{x}$ is the unitary operator for AE [8].

In contrast, HOE employs an n-qubit circuit that captures non-linear correlations within the state. Each component of the classical state $s(t)$ is normalized using $\tanh(x)$, and then encoded by the unitary operator $\mathbf{U}_{\mathrm{HOE}}^{x}$ as $|s_q(t)\rangle = \mathbf{U}_{\mathrm{HOE}}^{x}|0\rangle^{\otimes n}$. This unitary operator can be expressed as $\mathbf{U}_{\mathrm{HOE}}^{x} = \left(\prod_{i=0}^{n-2} \mathbf{U}_{\mathrm{ent}}\left(\tanh(x_i), \tanh(x_{i+1})\right)\right) \cdot \left(\prod_{i=0}^{n-1} \mathbf{RZ}\left(2\pi \tanh(x_i)\right) \cdot \mathbf{H}^{\otimes n}\right)$ [8].

After encoding, we utilize two parameterized quantum circuits (PQCs) to process the encoded quantum state $|s_q(t)\rangle$. One PQC is used for actor network to learn the policy, and other PQC is used for the critic network to estimate the value function. Both PQCs share the same architectural design, consisting of L layers and input dimension is similar to the encoding dimension. However, both are differ in their trainable

parameters. Thus, the unitary operator for PQC can be defined as $\mathbf{U}(\theta, \phi) = \prod_{l=1}^{L} \mathbf{U}_{\text{layer}}^{(l)}(\theta^{(l)}, \phi^{(l)})$, where $\mathbf{U}_{\text{layer}}^{(l)}(\theta^{(l)}, \phi^{(l)})$ is the unitary operation for the l-th layer which can be expressed as

$$\mathbf{U}_{\text{layer}}(\theta^{(l)}, \phi^{(l)}) = \left(\bigotimes_{i=0}^{n-1} \mathbf{RY}(\theta_i^{(l)}) \mathbf{RZ}(\phi_i^{(l)}) \right) \left(\prod_{i=0}^{n-2} \mathbf{CZ}_{i,i+1} \right), \tag{15}$$

where $\mathbf{RY}(\theta_i^{(l)})$ and $\mathbf{RZ}(\phi_i^{(l)})$ are single qubit rotation gates applied to qubit i in the l-th layer. Moreover, $\mathbf{CZ}_{i,i+1}$ introduce entanglement between adjacent qubits i and $i+1$. For the actor PQC, the unitary operator is $\mathbf{U}_{\text{actor}}(\theta_a, \phi_a)$, with trainable parameters $\theta_a = \{\theta_{a,i}^{(l)}\}$ and $\phi_a = \{\phi_{a,i}^{(l)}\}$. For the critic PQC, the unitary operator is $\mathbf{U}_{\text{critic}}(\theta_c, \phi_c)$, with trainable parameters $\theta_c = \{\theta_{c,i}^{(l)}\}$ and $\phi_c = \{\phi_{c,i}^{(l)}\}$.

When the actor PQC $\mathbf{U}_{\text{actor}}(\theta_a, \phi_a)$ is applied to the encoded quantum state $|s_q(t)\rangle$, we measure the policy using projective measurements in the Pauli-$\mathbf{Z}$ basis, where the expectation values $\langle \pi_{\theta_a,\phi_a} \rangle = \{\langle \mathbf{Z}_i \rangle\}_{i=0}^{n-1}$ are obtained. These expectation values are averaged over N_{shot} measurements and decoded through $F_{\text{actor}}^{\text{decode}}$ to compute the policy as

$$\pi_{\theta_a,\phi_a}(a(t) \mid s(t)) \xleftarrow{F_{\text{actor}}^{\text{decode}}} \frac{1}{N_{\text{shot}}} \sum_{l=1}^{N_{\text{shot}}} \mathbf{M}\left(\langle \pi_{\theta_a,\phi_a} \rangle\right). \tag{16}$$

The decoding operation transforms these expectation values into the mean of a normal distribution $\sim \mathcal{N}(\mu, \sigma)$ using a linear layer followed by a tanh activation, with the standard deviation $\sigma = \exp(\sigma_{\log})$. Similarly, the critic PQC $\mathbf{U}_{\text{critic}}(\theta_c, \phi_c)$ is applied to $|s_q(t)\rangle$, and we measure the value function using the same Pauli-$\mathbf{Z}$ basis. After sampling an action $a(t)$ from the policy $\pi_{\theta_a,\phi_a}(a(t) \mid s(t))$ and executing it in the SAGIN environment, the agent receives a reward $r(t)$, next state $s(t+1)$. Then, the TD target $y(t)$ can be calculated as [10]

$$y(t) = r(t) + \gamma \mathrm{V}_{\theta_c,\phi_c}(s(t+1)), \tag{17}$$

where γ is the discount factor and $\mathrm{V}_{\theta_c,\phi_c}$ is the critic PQC's estimated value of the next state $s(t+1)$. Then, the advantage value which quantifies how much better the action $a(t)$ can be calculated as [10]

$$\mathrm{A}(t) = r(t) + \gamma \mathrm{V}_{\theta_c,\phi_c}(s(t+1)) - \mathrm{V}_{\theta_c,\phi_c}(s(t)). \tag{18}$$

For the policy gradient approach, the actor loss can be defined as $L_{\text{actor}} = \mathbb{E}\left[-\log \pi_{\theta_a,\phi_a}(a(t) \mid s(t)) \mathrm{A}(t) - \mathrm{C}_e \mathrm{H}(\pi_{\theta_a,\phi_a})\right]$, where $\mathrm{H}(\pi_{\theta_a,\phi_a})$ is the entropy of the policy and C_e is the entropy coefficient [10]. Then, critic loss can be calculated as $L_{\text{critic}} = \mathbb{E}\left[\mathrm{C}_v \left(\mathrm{V}_{\theta_c,\phi_c}(s(t)) - y(t)\right)^2\right]$, where C_v is the value loss coefficient. These loss functions are then used to update the actor and critic parameters using the parameter-shift rule [8]. The detailed proposed QEA2C framework is in Algorithm 1.

Algorithm 1 Proposed QEA2C Algorithm for Solving (14)

Initialize:
  Set up the SAGIN environment with specified parameters.
  Initialize the actor PQC $\mathbf{U}_{\text{actor}}(\theta_a, \phi_a)$ with parameters θ_a, ϕ_a.
  Initialize the critic PQC $\mathbf{U}_{\text{critic}}(\theta_c, \phi_c)$ with parameters θ_c, ϕ_c.
  Set the Adam optimizer with learning rates and other hyperparameters.
for episode = 1 to E **do**
  Reset environment to obtain initial state $s(t)$.
  Initialize time step $t = 0$.
  while $t < T$ and not done **do**
    Encode state $s(t)$ into $|s_q(t)\rangle$.
    Apply actor PQC: $\mathbf{U}_{\text{actor}}(\theta_a, \phi_a)|s_q(t)\rangle$, measure in Pauli-$\mathbf{Z}$ basis and decode to get $\pi_{\theta_a,\phi_a}(a(t) \mid s(t))$.
    Sample action $a(t)$ from policy $\pi_{\theta_a,\phi_a}(a(t) \mid s(t))$.
    Execute $a(t)$ in SAGIN; receive $\{r(t), s(t+1), d(t)\}$.
    Apply critic PQC on $s(t)$: $\mathbf{U}_{\text{critic}}(\theta_c, \phi_c)|s_q(t)\rangle$, measure in Pauli-$\mathbf{Z}$ basis, and decode to get $\mathrm{V}_{\theta_c,\phi_c}(s(t))$.
    Encode state $s(t+1)$ into $|s_q(t+1)\rangle$.
    Apply critic PQC on $s(t+1)$: $\mathbf{U}_{\text{critic}}(\theta_c, \phi_c)|s_q(t+1)\rangle$, measure in Pauli-$\mathbf{Z}$ basis, and decode to get $\mathrm{V}_{\theta_c,\phi_c}(s(t+1))$.
    Compute TD target $y(t)$ using (17).
    Calculate the advantage $\mathrm{A}(t)$ using (18).
    Update actor parameters: $\theta_a, \phi_a \leftarrow \theta_a, \phi_a + \alpha_\theta \nabla_{\theta_a,\phi_a} L_{\text{actor}}$.
    Update critic parameters: $\theta_c, \phi_c \leftarrow \theta_c, \phi_c + \alpha_\phi \nabla_{\theta_c,\phi_c} L_{\text{critic}}$.
    Update state $s(t) \leftarrow s(t+1)$; increment $t \leftarrow t+1$.
    If $d(t) = 1$, **break**.
  end while
end for

1) Complexity Analysis: The computational complexity of the QEA2C algorithm per step depends on the encoding method. AE costs $O(n)$, with PQC operations at $O((3m-1)L \cdot 2^m)$, measurements $O(mN_{\text{shot}})$, post-processing $O(mn)$, and gradients dominating at $O(mL^2(3m-1) \cdot 2^m)$. Therefore, total complexity can be expressed as $O(mL^2(3m-1) \cdot 2^m + mN_{\text{shot}} + mn)$. HOE costs $O(n \cdot 2^n)$, with PQC operations at $O((3n-1)L \cdot 2^n)$, measurements $O(nN_{\text{shot}})$, post-processing $O(n^2)$, and gradients at $O(nL^2(3n-1) \cdot 2^n)$. Therefore, total complexity can be calculated as $O(nL^2(3n-1) \cdot 2^n + n \cdot 2^n + nN_{\text{shot}} + n^2)$. Overall, HOE is more complex due to the increased number of qubits.

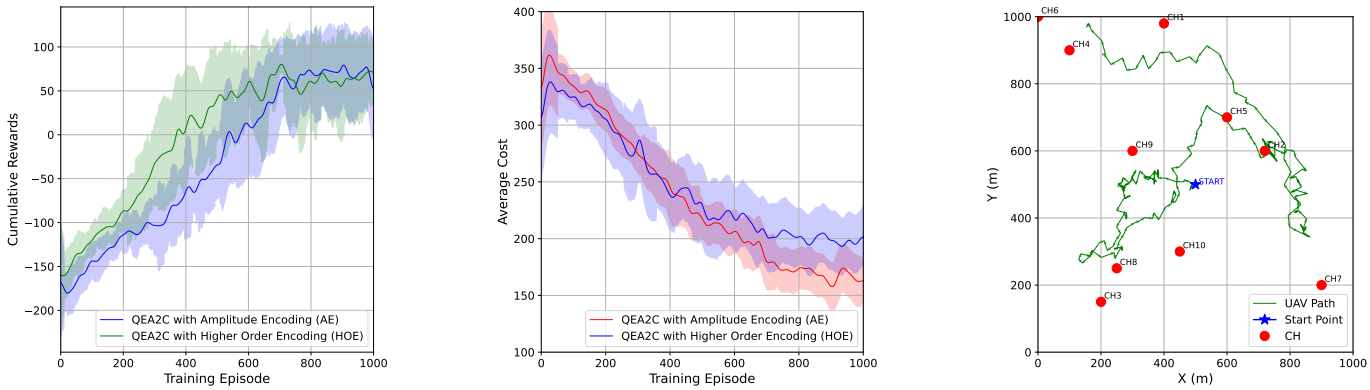
IV. Numerical Results And Discussions

A. Simulation Settings

TABLE I: Simulation parameters [1], [3], [4], [5], [9]

Parameter	Value
Number of CHs, M	10 (Random Positions)
Number of antennas at UAV, K	8
$x^{\max}$, $y^{\max}$	(1000, 1000) m
Distance from Earth surface to LEO, R	300 km
UAV MEC computation power, f_{m}^{u}	2 GHz
Maximum UAV speed, $V^{\max}$	30 m/s
SNSs MEC computation power, $f_{\text{m}}^{\text{ss(i)}}$	1 GHz
Task size, D_{m}	$[5 \times 10^5 - 2 \times 10^6]$ bits
Task complexity, Q_{m}	$[5 \times 10^5 - 2 \times 10^6]$ cycles
Transmit power of UAV, MNS p_{u}, $p_{\text{ms,ss(i)}}$	5 W
System bandwidth, $\mathcal{B}$	10 MHz
Weighting factors, w_1, w_2, w_3	0.02, 0.5, 0.5

This subsection presents the settings of parameters for the implementation of the proposed solution and simulations. The parameters of the QEA2C algorithm are as follows. The actor and critic learning rates are both set to 1×10^{-4}, and the discount factor γ is 0.99. The value loss coefficient C_v is 0.5, while the entropy coefficient C_e is 0.01. The training



(a) Convergence Performance (b) Average Cost with Training Episodes (c) UAV's Trajectory with HOE

Fig. 2: Analysis of system performance across (a) convergence, (b) average cost, and (c) UAV trajectory with HOE.

process is carried over 1000 episodes, each episode compose with 200 steps T. For AE, 3 qubits are used, whereas HOE employs 8 qubits. The quantum circuit comprises 4 layers L and uses 1024 measurement shots N_{shot}. Simulations were carried out using `TorchQuantum` library [7]. Furthermore, SAGIN environment parameters are summarized in TABLE I.

B. Numerical Results

1) Training performance: As shown in Fig. 2a, we compare the cumulative rewards for QEA2C using AE and HOE over 1000 training episodes. While both methods illustrates an upward trend in performance. However, the HOE ultimately achieves a higher cumulative reward and shows faster convergence. These results shows that the choice of encoding scheme plays a crucial role in learning efficiency and offers clear advantages in accelerating convergence.

2) Average cost performance: In Fig. 2b, the AE consistently achieves a lower average cost over the training period compared to HOE. Although HOE shows a steeper decline in cost at the beginning, its performance eventually stabilizes at a higher cost level. This indicates that AE delivers a more reliable reduction in average cost over time. Furthermore, these observations underscore the importance of selecting an encoding method that aligns with cost minimization objectives.

3) UAV's trajectory: Fig. 2c illustrates the UAV's trajectory under the QEA2C algorithm using HOE. The UAV starts from a central location and follows a dynamic path to serve the all CHs distributed across the area. Here the Chs are generated randomly. Thus, the path reflects the agent's ability to make adaptive decisions based on task status, location, and system constraints. With HOE, the UAV demonstrates flexible movement and effective coverage, adjusting its route to optimize task handling while minimizing cost. The trajectory also indicates efficient area exploration and strategic selection of CHs throughout the mission.

V. Conclusion

This paper introduced a 6G-enabled SAGIN framework integrating LEO nano satellite constellations and a UAV with MEC capabilities. We formulated a cost minimization problem involving UAV trajectory, task offloading, and satellite load distribution as a MINLP problem, and solved it using the QEA2C framework, which incorporates AE and HOE techniques. By comparing two quantum encoding schemes, this study shows clear differences in their learning behaviors and system impact. HOE proved beneficial for fast policy learning and reliable CH coverage due to its expressive state representation. On the other hand, AE maintained a clear advantage in energy and cost efficiency, making it a better fit where resource constraints are critical. These results highlight the trade off between performance and complexity in quantum state encoding. In future work, we plan to explore adaptive encoding strategies and hybrid quantum-classical learning that can adaptively switch modes, further enhancing flexibility in large-scale, real-time 6G-enbled SAGINs.

Acknowledgments

This work was supported in part by the Canada Excellence Research Chair Program CERC-2022-00109, by the NSERC Discovery Grant Program RGPIN-2025-04941, and by the Canada First Research Excellence Fund (CFREF) for the project Transforming Climate Action (TCA).

References

[1] G. S. Kim, Y. Cho, S. Park, S. Jung, and J. Kim, "Quantum multi-agent reinforcement learning for joint cube-satellites and high-altitude long-endurance aerial vehicles in SAGIN," *IEEE Trans. Aerosp. Electron. Syst.*, Apr. 2025, doi:10.1109/TAES.2025.3556050.

[2] T. T. Bui *et al.*, "Impact of 6G space-air-ground integrated networks on hard-to-reach areas: Tourism, agriculture, education, and indigenous communities," *EAI Endorsed Trans. on Tourism, Tech. and Intelli*, vol. 1, no. 1, pp. 1–8, Sep. 2024.

[3] M.-H. T. Nguyen *et al.*, "Real-time optimized clustering and caching for 6G satellite-UAV-terrestrial networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 3, pp. 3009–3019, Mar. 2024.

[4] G. Sun, W. Ma, J. Li, Z. Sun, J. Wang, D. Niyato, and S. Mao, "Task delay and energy consumption minimization for low-altitude MEC via evolutionary multi-objective deep reinforcement learning," *arXiv preprint arXiv:2501.06410*, Jan. 2025.

[5] D. V. Huynh *et al.*, "Joint sensing, communications, and computing design for 6G URLLC service-oriented MEC networks," *IEEE Internet of Things J.*, vol. 11, no. 20, pp. 32 429–32 439, October 2024.

[6] B. Narottama and S. Y. Shin, "Quantum neural networks for resource allocation in wireless communications," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 1103–1115, Feb. 2022.

[7] H. Wang, *et al.*, "Quantumnas: Noise-adaptive search for robust quantum circuits," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2022.

[8] M. Monnet, N. Chaabani, T.-A. Drăgan, B. Schachtner, and J. M. Lorenz, "Understanding the effects of data encoding on quantum-classical convolutional neural networks," *arXiv preprint arXiv:2405.03027*, May 2024.

[9] Silvirianti, B. Narottama, and S. Y. Shin, "UAV coverage path planning with quantum-based recurrent deep deterministic policy gradient," *IEEE Trans. Veh. Technol.*, vol. 73, no. 5, pp. 7424–7429, May 2024.

[10] K. Zhu, S. Li, X. Zhang, J. Wang, C. Xie, F. Wu, and R. Xie, "An energy-efficient dynamic offloading algorithm for edge computing based on deep reinforcement learning," *IEEE Access*, vol. 12, pp. 127 489–127 504, Sep. 2024.