DRL-based Optimisation for Task Offloading in Space-Air-Ground Integrated Networks: A Reliability-Driven Approach

Dang Van Huynh*, Saeed R. Khosravirad†, Simon L. Cotton‡ Octavia A. Dobre*, Trung Q. Duong*,‡
* Memorial University, Canada, e-mail:{vdhuynh, odobre, tduong}@mun.ca
† Nokia Bell Labs, USA, e-mail: saeed.khosravirad@nokia-bell-labs.com
‡ Queen's University Belfast, UK e-mail:{simon.cotton, trung.q.duong}@qub.ac.uk

***Abstract*—This paper addresses the problem of reliable task offloading in space-air-ground integrated network (SAGIN) based edge computing systems. Specifically, we aim to maximise the successful task offloading ratio for ground users communicating with a satellite's edge server. In our network topology, end-to-end communications are facilitated by relay unmanned aerial vehicles (UAVs). The formulated problem jointly optimises task offloading portions and bandwidth allocations for both ground-to-air and air-to-space links, subject to quality-of-service (QoS) requirements, transmission rates, system bandwidth, and the computing capacity of the satellite's edge server. To solve the formulated complex *non-linear*, *non-convex*, and *mixed-integer* problem, we propose an efficient solution underpinned by a deep reinforcement learning (DRL). Simulation results demonstrate the effectiveness of the proposed method, which achieves stable training performance and an optimised reliable offloading ratio compared to benchmark schemes.**

I. Introduction

Space-air-ground integrated networks (SAGIN) are emerging as a key technology for achieving ubiquitous connectivity in 6G networks [1], [2]. By integrating space, aerial, and terrestrial components, SAGIN can provide seamless wireless coverage across vast geographical areas, including hard-to-reach locations, making it essential for critical services such as remote surveillance, environmental monitoring, and disaster management [2]. However, SAGIN presents several challenges, which have attracted considerable attention from the research community. One such challenge lies in optimising resource management across both communication resources (e.g., bandwidth allocation, transmission power) and computing resources (e.g., processing capacity, storage, energy) [3], [4]. The high attenuation and long-distance nature of satellite communications, combined with the resource limitations of ground devices used for remote operation, make the design of efficient solutions for SAGIN-assisted systems particularly complex [2]. Addressing these challenges will be crucial for fully realising the potential of SAGIN in real-world applications [5], [6].

Recently, the integration of SAGIN-based communication with edge and cloud computing, driven by the need to support emerging services that demand low latency and high computational power, has gained significant attention [7]–[15]. The convergence of these key technologies will unlock the full potential of next-generation wireless networks, providing the ability to not only deliver global coverage but at the same time enhance computational capacity to meet complex service demands. Satellites equipped with edge servers to process computational tasks offloaded from ground users are a key element of this integration. These satellites, with their powerful computing resources, are able to process complex tasks and provide timely responses to users on the ground, greatly reducing latency compared to routing from remote locations via ground based telecommunications infrastructure, which in some cases may not be available.

The benefits of using SAGIN assisted edge computing are not open-ended. A major challenge here is long-distance transmissions between satellites and ground users which presents considerable challenges in maintaining efficient communication, making joint optimisation of communication and computing resources a difficult and multi-faceted problem. The dynamic nature of user demands, network conditions, and resource availability requires sophisticated strategies for balancing these resources. For example, service deployment and task scheduling are essential to improving network service capabilities while also reducing deployment and operational costs [7]. Similarly, the joint selection of servers and services plays a crucial role in achieving optimal configurations for computing services across the SAGIN infrastructure [11]. Common objectives in this field include minimising energy consumption [9], [14], reducing latency [15], and maximising resource utilisation efficiency [10]. A range of optimisation methods has been proposed to address these challenges, including both traditional mathematical optimisation techniques [8], [10], [11] and more modern machine learning approaches [12], [15], [16]. Among these, machine learning methods, particularly deep reinforcement learning (DRL), have been widely adopted due to their ability to adapt to changing network conditions and user demands in real time [7], [9], [15]. DRL, for example, enables the system to learn optimal policies for resource management by interacting with the environment, offering a promising solution for addressing the dynamic and complex nature of SAGIN systems.

While various research objectives have been explored in SAGIN-based systems, a critical challenge remains: the reliable offloading and processing of tasks, particularly in remote areas and critical scenarios such as emergency responses and disaster recovery. In these situations, timely and efficient task execution can be vital, making the issue of reliable task offloading a key priority. Motivated by these challenges, this paper focuses on the problem of reliable task offloading

in SAGIN-assisted edge computing. We aim to maximise the ratio of successfully offloaded tasks in SAGIN systems, ensuring that tasks are completed with the required quality of service (QoS) and within the resource limitations. The proposed solution jointly optimises task offloading portions and bandwidth allocation for transmissions between ground users, relay UAVs, and the satellite, while considering QoS requirements and resource constraints. By addressing these aspects, our approach provides a robust framework for improving task reliability and overall system performance in critical and remote scenarios.

II. System Model and Problem Formulation

In this paper, we consider a SAGIN model which consists of N ground users (GUs), M relay UAVs, and one satellite associated with an edge server to process offloaded tasks from the GUs. Fig. 1 provides an illustration of the considered system model. We assume that the formation of GU-UAV networks is conducted in advanced, where the j-th UAV only serves a finite number of GUs in its coverage. A computational task generated from the i-th GU is characterised by a tuple of three parameters (S_i, C_i, D_i), denoting the task size (bits), required CPU cycles (cycles), and delay tolerance (seconds), respectively. Due to limitation in the available computing capacity, as well as energy budget, the GUs have to offload the task to the satellite's edge server to process.

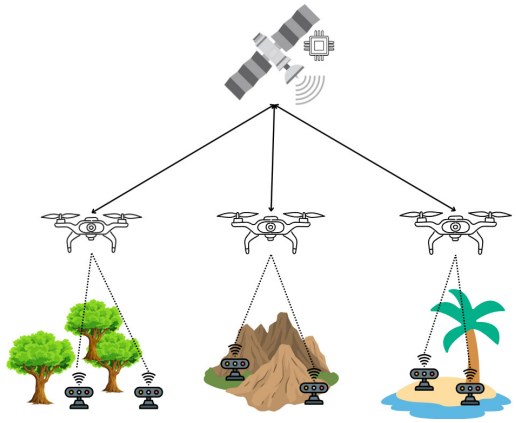

Fig. 1. An illustration of SAGIN-enabled edge computing systems.

A. Wireless Transmission Model

In this paper, all devices considered in the system model are single-antenna devices. Frequency division multiple access (FDMA) method is utilised for wireless transmissions in the system. We aim to develop an optimal design for bandwidth allocations, which guarantees the transmission QoS, meets the desired latency constraints, and improves the reliability of task offloading.

1) Channel model for space-to-air transmissions: We adopt the shadowed-Rician fading (SRF) model to describe the channel of space-to-air transmissions [17]. The channel gain h_j^{AS} is expressed as $h_j^{\mathsf{AS}} = \sqrt{(d_0/d_j)^{\alpha}}\mathrm{SR}(\omega, \delta, \epsilon)$. Here d_j is the distance between the j-th UAV and the satellite; $d_0 = 1$ is a reference distance and α is the path loss exponent for the space-to-air link; ω is the average power of the direct LoS component; δ is the half average power of the scatter portion; ϵ is the Nakagami m-parameter for the scattered NLoS components.

2) Channel model for air-to-ground transmissions: In this work, we consider light-of-sight (LOS) links between the i-th GU and j-th UAVs so we can model the channel gain $h_{i,j}^{\mathsf{GA}}$ as $h_{i,j}^{\mathsf{GA}} = \sqrt{\beta_{i,j}(d_{i,j})}g_{i,j}$ [9]. Here, $\beta_{i,j}(d_{i,j}) = d_0/d_{i,j}^{\alpha}$ represents the large-scale fading, including distance-based path loss and shadowing, where α and $d_0 = 1$ m are the path loss exponent for the air-to-ground links and the reference distance, respectively. $g_{i,j} \sim \text{Rician}(K)$ is the small-scale fading component, where K is the Rician factor defining the ratio of power of the direct LoS path to the power contributed by the scattered paths.

3) Transmission schemes: The transmission rate of the i-th GU to the j-th UAV is given by

$$r_{i,j}^{\mathsf{GA}}\left(b_{i,j}^{\mathsf{GA}}\right) = b_{i,j}^{\mathsf{GA}}\log_2\left(1 + \frac{P_i h_{i,j}^{\mathsf{GA}}}{N_0 b_{i,j}^{\mathsf{GA}}}\right). \tag{1}$$

Similarly, the transmission rate of the j-th UAV to the satellite is expressed as

$$r_j^{\mathsf{AS}}\left(b_j^{\mathsf{AS}}\right) = b_j^{\mathsf{AS}}\log_2\left(1 + \frac{P_j h_j^{\mathsf{AS}}}{N_0 b_j^{\mathsf{AS}}}\right). \tag{2}$$

where $r_{i,j}^{\mathsf{GA}}$ denotes the transmission rate of the i-th GU to the j-th UAV, while r_j^{AS} is the transmission rate from the j-th UAV to the satellite. Here, $b_{i,j}^{\mathsf{GA}}$ is bandwidth allocated for the link from the i-th GU to the j-th UAV, and b_j^{AS} is bandwidth allocated for the link from the j-th UAV to the satellite. P_i and P_j are the transmission power of the i-th GU and the j-th UAV, respectively. $h_{i,j}^{\mathsf{GA}}$ is channel gain between the i-th GU and the j-th UAV, and h_j^{AS} is the channel gain between the UAV and the satellite. N_0 is the noise spectral density.

B. Latency Model

As illustrated in Fig. 1, the i-th GU offloads a portion of $x_{i,j}$ of the computational task to the j-th UAV, then the UAV forwards it to the satellite's edge server for processing. Therefore, the latency of the i-th task consists of four components: local processing latency $\left(L_i^{\mathsf{GU}}\right)$, GU-to-UAV transmission latency $\left(L_{i,j}^{\mathsf{GA}}\right)$, UAV-to-satellite transmission latency $\left(L_j^{\mathsf{AS}}\right)$, and edge processing latency $\left(L_i^{\mathsf{ES}}\right)$, which calculated as follows.

$$L_i^{\mathsf{GU}}(x_{i,j}) = \frac{(1 - x_{i,j})C_i}{f_i^{\mathsf{GU}}}, \tag{3}$$

$$L_{i,j}^{\mathsf{GA}}(x_{i,j}, b_{i,j}^{\mathsf{GA}}) = \frac{x_{i,j}S_i}{r_{i,j}^{\mathsf{GA}}\left(b_{i,j}^{\mathsf{GA}}\right)}, \tag{4}$$

$$L_{i,j}^{\mathsf{AS}}(x_{i,j}, b_j^{\mathsf{AS}}) = \frac{x_{i,j}S_i}{r_j^{\mathsf{AS}}\left(b_j^{\mathsf{AS}}\right)}, \tag{5}$$

$$L_i^{\mathsf{ES}}(x_{i,j}) = \frac{x_{i,j}C_i}{f_i^{\mathsf{ES}}}. \tag{6}$$

As a result, the total latency for a task completely offloaded

and processed is given by

$$L_i = L_i^{\mathsf{GU}} + L_{i,j}^{\mathsf{GA}} + L_{i,j}^{\mathsf{AS}} + L_i^{\mathsf{ES}}. \tag{7}$$

C. Energy Consumption Model

To handle the limitation on the energy budget of the GUs, we model the energy consumption of the i-th GU as follows

$$E_i\left(x_{i,j}, b_{i,j}^{\mathsf{GA}}\right) = \theta(1 - x_{i,j}) C_i \left(f_i^{\mathsf{GU}}\right)^2 + \frac{x_{i,j} S_i P_i}{r_{i,j}^{\mathsf{GA}}}, \tag{8}$$

which includes two components: energy consumption for local processing and energy consumption for the transmission. Since the task is partially offloaded a portion of $x_{i,j}$ to the UAVs, the i-th GU only processes the remaining portion of $(1 - x_{i,j})$. Here, θ is the parameter used to calculate the computation energy of GUs, which varies according to the CPU used [18].

D. Reliable Task Offloading Definition

In this paper, we propose a reliability-driven approach for optimal design of joint task offloading and bandwidth allocations in SAGIN-assisted edge computing. The reliable metric is developed based on a binary indicator $\phi_{i,j} = \{0, 1\}$. Specifically, a task is considered reliably offloaded if the total latency L_i is less than or equal to its delay tolerance D_i, mathematically expressed as

$$\phi_{i,j}\left(x_{i,j}, b_{i,j}^{\mathsf{GA}}, b_j^{\mathsf{AS}}\right) = \begin{cases} 1, & \text{if } L_{i,j} \leq D_i, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

E. Problem Formulation

Based on the above representation of the system model, the optimisation problem formulated in this paper is given by (10). Here, the objective of the problem is to maximise the average reliable task offloading ratio, ensuring the tasks are completely offloaded and processed within their delay tolerances by optimising the variables of offloading portions, i.e., $\mathbf{x} \triangleq \{x_{i,j}\}_{\forall i,j}$ and bandwidth allocations, i.e., $\mathbf{b} \triangleq \{b_{i,j}^{\mathsf{GA}}, b_j^{\mathsf{AS}}\}_{\forall i,j}$.

$$\textbf{P1:} \underset{\mathbf{x},\mathbf{b}}{\text{maximise}} \ \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} \phi_{i,j}\left(x_{i,j}, b_{i,j}^{\mathsf{GA}}, b_j^{\mathsf{AS}}\right) x_{i,j}, \tag{10a}$$

$$\text{s.t. } E_i\left(x_{i,j}, b_{i,j}^{\mathsf{GA}}\right) \leq E_i^{\max}, \forall i, \tag{10b}$$

$$\sum_{i=1}^{N} b_{i,j} \leq B_j^{\max}, \forall j, \tag{10c}$$

$$\sum_{j=1}^{M} b_j \leq B^{\mathsf{SAT}}, \tag{10d}$$

$$r_{i,j}^{\mathsf{GA}} \geq r_{i,j}^{\min}, \forall i, j, \tag{10e}$$

$$r_j^{\mathsf{AS}} \geq r_j^{\min}, \forall j, \tag{10f}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{M} x_{i,j} f_i^{\mathsf{ES}} \leq F^{\max}. \tag{10g}$$

In (10), (10b) represents the constraint of the energy budget of the GUs. Constraints (10c) and (10d) are constraints for the bandwidth allocations of GU-to-UAV links and UAV-to-satellite links, respectively. Constraints (10e) and (10f) are the QoS requirements for the transmission rates. Lastly, constraint (10g) guarantees the computing capacity of the satellite's edge server against exceeding maximum setting.

III. Proposed Solution

It is obvious that the problem given in (10) comprises of non-linearities, non-convexity, coupled constraints, and binary indicators in the objective function, which make the problem challenging for classical optimisation methods to find optimal solutions effectively. In contrast, DRL offers a flexible, scaleable, and adaptive approach to learning optimal policies in dynamic environments, making it an attractive solution for solving the presented problem. By leveraging exploration and function approximation, DRL can find near-optimal solutions that meet the problem's constraints while maximising the task offloading ratio. Therefore, we propose a DRL-based optimisation solution to tackle the formulated problem. More specifically, the optimisation variables of the problem include the task offloading portions and the bandwidth allocations, which are all continuous variables. Consequently, the deep deterministic policy gradient (DDPG) algorithm is selected to develop the solution for this paper.

A. Reinforcement Learning Representation

We are in the position of transforming the original problem (10) into a problem that can be solved by DRL-based algorithms. To solve a problem with DRL algorithms, the optimisation problem needs to be reformulated as a Markov decision process (MDP) formulation, including state space ($\mathcal{S}$), action space ($\mathcal{A}$), and the reward function ($\mathcal{R}$). We first start with the design of the state space.

1) State space: The state space $\mathcal{S}$ is composed of necessary information of the system at the state t, observed by the agent to select next action, including the following system parameters:

- Task size $S_i(t)$: The size of the computational task generated by the i-th GU, measured in bits.
- Required CPU cycles $C_i(t)$: The number of CPU cycles required to process the task generated by the i-th GU, measured in cycles/second.
- Delay tolerance $D_i(t)$: The maximum allowable latency for the task generated by the i-th GU, measured in seconds.
- Bandwidth allocations $b_{i,j}^{\mathsf{GA}}(t)$ and $b_j^{\mathsf{AS}}(t)$: The bandwidth allocated to the i-th GU for communication with the j-th UAV and for the communication of the j-th UAV with the satellite, respectively.
- Channel conditions $h_{i,j}^{\mathsf{GA}}(t)$ and $h_j^{\mathsf{AS}}(t)$:The wireless channel gains from the i-th GU to the j-th UAV, and from the j-th UAV and the satellite, respectively.
- Energy consumption $E_i(t)$: The current energy consumption of the i-th GU, measured in joules.

It is important to note that the DDPG agent can learn more effectively with a concentrated state space, instead of discrete information. The concentrated state space can incorporate constraint violations as part of the state, making it

easier for the agent to learn feasible solutions, focusing on optimising the key metrics that matter. Therefore, we propose the concentrated expression of the state space $\mathcal{S}$ as follows

$$\mathbf{s_t} = \{R_{\text{off}}(t), V_{\text{con}}(t), U_{\text{util}}(t)\}, \tag{11}$$

where:

- $R_{\text{off}}(t)$ is the task completion ratio at time t, indicating the percentage of tasks completed within the allowed delay;
- $V_{\text{con}}(t)$ represents the number of system constraint violations up to time t, with penalties applied according to $\boldsymbol{\lambda}$ for each violation;
- $U_{\text{proc}}(t)$ is the utilisation of the computing capacity at the satellite's edge server, calculated as:

$$U_{\text{proc}}(t) = \frac{\sum_{i=1}^{N} x_{i,j} f_i^{\mathsf{ES}}}{F^{\max}}, \tag{12}$$

where $x_{i,j}$ is the offloading portion, f_i^{ES} is the allocated processing rate for task offloaded from the i-th GU, and $F^{\max}$ is the maximum computing capacity of the satellite's edge server.

2) Action space: The action space $\mathcal{A}$ consists of the following decisions made by the agent:

- Offloading portion $x_{i,j}(t)$: The portion of the task offloaded from the i-th GU to the j-th UAV;
- Bandwidth allocation $b_{i,j}^{\mathsf{GA}}(t)$: The bandwidth allocated to the i-th GU for communication with the j-th UAV;
- Satellite bandwidth allocation $b_j^{\mathsf{AS}}(t)$: The bandwidth allocated to the j-th UAV for air-to-space communications.

Thus, the action space $\mathcal{A}$ is represented as

$$\mathcal{A} = \{x_{i,j}(t), b_{i,j}^{\mathsf{GA}}(t), b_j^{\mathsf{AS}}(t)\}_{\forall i,j}. \tag{13}$$

where $x_{i,j} \in [0,1]$ represents the offloading portion, and $b_{i,j} \in [0, B_j^{\max}]$, $b_j \in [0, B^{\mathsf{SAT}}]$ represents the bandwidth allocations.

3) Reward function: The reward r_t at time step t is designed to encourage efficient task offloading, minimise latency, and reduce energy consumption. The reward is calculated as

$$r_t = \sum_{i=1}^{N}\sum_{j=1}^{M} \left(\delta_{i,j} x_{i,j} - \lambda_{\text{E}}\Psi_{\text{E}} - \lambda_{\text{B}}\Psi_{\text{B}} - \lambda_{\text{r}}\Psi_{\text{r}} - \lambda_{\text{F}}\Psi_{\text{F}}\right), \tag{14}$$

where:

- $\delta_{i,j}$ is an indicator function that equals 1 if the total task latency $L_{i,j} \leq D_i$, and 0 otherwise, defined in (9);
- $x_{i,j}$ is the offloaded portion generated by the i-th GU;
- $\lambda_{\text{E}}, \lambda_{\text{B}}, \lambda_{\text{r}}$ and λ_{F} are introduced weighting factors for penalising energy consumption, bandwidth budget, minimum transmission rates, and the satellite's computing budget, respectively;
- Ψ_{E}, Ψ_{B}, Ψ_{r}, Ψ_{E} present how much the constraints in (10) are violated.

By designing the reward function in this way, the agent is encouraged to maximise the reliable task offloading portion while penalising high energy consumption, task latency, exceeding bandwidth budget and computing capacity, thereby efficiently finding the optimal solution for the original problem (10).

B. Implementation of the Proposed DDPG-based Solution

The proposed solution is constructed from the DDPG algorithm, implemented with the framework of actor-critic networks. The actor network is responsible for mapping the current state of the environment to a continuous action, which is fully connected and consists of three layers: the input layer, hidden layer, and output layer. The hidden layer in this network works as an approximator for the policy function. On the other hand, the critic network evaluates how good a particular action is for a given state by estimating the Q-value (i.e., the expected cumulative reward), expressed as (15) [19]. It takes both the current state and the action as input, and outputs a scalar value.

$$Q^{\pi}(s_t, a_t) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \,\middle|\, s_t = s, a_t = a\right]. \tag{15}$$

Thus, the Q-value $Q(s_t, a_t)$ represents the expected cumulative reward for taking action a_t in state s_t, considering the agent's future states and actions under the current policy.

During the training process, the Q-value is updated in DDPG using the Bellman equation:

$$y_t = r_t + \gamma Q'(s_{t+1}, \pi'(s_{t+1})|\theta^{Q'}), \tag{16}$$

where y_t is the target Q-value, the discount factor; γ is the discount factor, and $Q'(s_{t+1}, \pi'(s_{t+1})|\theta^{Q'})$ is the Q-value predicted by the *target critic network* for the next state s_{t+1} and action $\pi'(s_{t+1})$, using the *target actor network* π'. The critic network is trained by minimising the loss between the predicted Q-value $Q(s_t, a_t|\theta^Q)$ and the target Q-value y_t:

$$\mathcal{L}(\theta^Q) = \mathbb{E}\left[\left(y_t - Q(s_t, a_t|\theta^Q)\right)^2\right]. \tag{17}$$

It is important to note that, in the implementation of DDPG algorithm, the replay buffer is a crucial component. The replay buffer works as a memory buffer that stores the agent's experiences from interacting with the environment. Each experience is stored in the relay buffer as a tuple (s_t, a_t, r_t, s_{t+1}). By sampling random mini-batches from the buffer, DDPG trains more effectively, reusing valuable experiences from previous interactions. In summary, the proposed DDPG-based algorithm for solving the problem formulated in (10) is provided in Algorithm 1.

IV. Simulation Results and Discussions

A. Parameter Settings

For simulations, we consider a system model that consists of $M = 3$ UAVs, $N = \{15, 21\}$ GUs. We assume that the assignments of UAVs and GUs are conducted in advance, with each UAV serving the same number of GUs, e.g., each UAV serves 5 GUs within its coverage area for the scenario where there are $N = 15$ GUs. The simulations are conducted in Python, making use of packages such as `Pytorch`,

Algorithm 1 : Proposed DDPG-based Algorithm for Solving **P1** (10).

Initialise actor network $\mu(s|\theta^\mu)$ and critic network $Q(s,a|\theta^Q)$ with random weights θ^μ and θ^Q;
Initialise target networks μ' and Q' with weights $\theta^{\mu'} \leftarrow \theta^\mu$, $\theta^{Q'} \leftarrow \theta^Q$;
Initialise replay buffer $\mathcal{R}$;
Initialise Ornstein-Uhlenbeck noise $\mathcal{O}$ for exploration;
for episode = 1 to E **do**
  Initialise a random process $\mathcal{O}$ for action exploration;
  Receive initial state s_1;
  for t = 1 to T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{O}_t$ (with noise for exploration);
    Execute action a_t and observe reward r_t and next state s_{t+1};
    Store transition (s_t, a_t, r_t, s_{t+1}) in replay buffer $\mathcal{R}$;
    Sample a random mini-batch of N_b transitions $(s_\ell, a_\ell, r_\ell, s_{\ell+1})$ from $\mathcal{R}$;
    Set $y_\ell = r_\ell + \gamma Q'(s_{\ell+1}, \mu'(s_{\ell+1}|\theta^{\mu'})|\theta^{Q'})$;
    Update critic by minimising the loss:

$$L = \frac{1}{N_b} \sum_{\forall \ell} \left(y_\ell - Q(s_\ell, a_\ell|\theta^Q)\right)^2;$$

    Update the actor using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N_b} \sum_{\forall i} \nabla_a Q(s,a|\theta^Q)|_{a=\mu(s)} \nabla_{\theta^\mu} \mu(s|\theta^\mu);$$

    Update target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'};$$
$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}.$$

  end for
end for

`gymnasium`, `pandas`, and `matplotlib` to implement the proposed solution and visualise numerical results.

For training the DDPG model, we set the learning rate of the actor to 10^{-5} while the learning rate of critic is 10^{-4}. The discount factor is set to $\gamma = 0.99$ and the factor for target network update is $\tau = 0.05$. The batch size for sampling in the training is set to 256 and the maximum size of the reply buffer is 10^6. Other communication and computing parameters are provided in Table I.

TABLE I
SIMULATION PARAMETERS [17], [18], [20].

Parameters	Value
Distance from UAVs to GUs	$d_{i,j} \sim \mathcal{U}(400, 500)$ m
Satellites' altitude	780 km
SRF model	$(\omega, \delta, \epsilon) = (5e^{-4}, 0.063, 2)$
Noise spectral density	$N_0 = -174$ dBm/Hz
Path-loss exponent	$\alpha = 2$
Rician K-factor	$K = 5$
Task size	$S_i \sim \mathcal{U}(100, 500)$ KB
Required CPU cycles of tasks	$C_i \sim \mathcal{U}(1000, 1200)$ megacycles.
Delay tolerance	$D_i \sim \mathcal{U}(2, 5)$ s
Maximum energy consumption of GU	$E_i^{\max} \sim \mathcal{U}(1, 1.5)$ J
Energy consumption parameter	$\theta = 10^{-27}$ Watt.s^3/cycle^3
Transmission power of GU	$P_i = 20$ dBm
Transmission power of UAV	$P_j = 37$ dBm
Maximum bandwidth for each UAV	$B_j^{\max} = 20$ MHz
Maximum bandwidth for the satellite	$B^{\text{SAT}} = 100$ MHz.

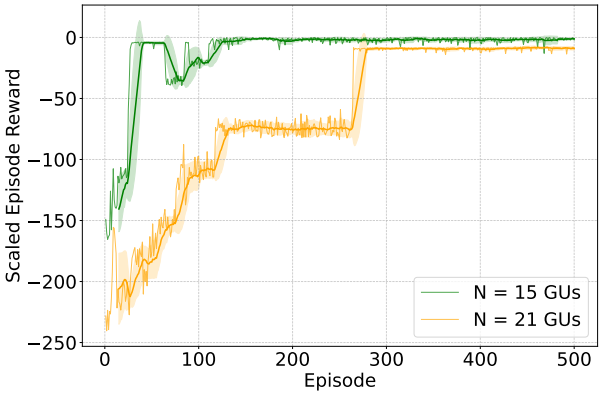

Fig. 2. Training performance over time of Algorithm 1 for the scenarios of $M = 3$ UAVs with $N = \{15, 25\}$ GUs.

B. Numerical Results

1) Training performance: The training performance of the proposed algorithm is displayed in Fig. 2, where the episode reward is plotted against the training episodes for two different scenarios $N = 15$ GUs and $N = 21$ GUs, both with $M = 3$ UAVs. The results show that in both scenarios, the algorithm demonstrates an upward trend in rewards as the training progresses, indicating successful learning. For $N = 15$ GUs, the algorithm converges faster, reaching a stable reward by around episode 100, with minimal fluctuations. In contrast, the case with $N = 21$ GUs exhibits a slower convergence rate, stabilising around episode 250. This difference in convergence speed can be attributed to the increased complexity of managing more users, which adds to the challenge of reliable task offloading. However, in both cases, the algorithm achieves stable and consistent rewards as the training progresses, highlighting its robustness and effectiveness in handling varying numbers of ground users. The shaded regions around the curves represent the standard deviation, showing that the variability in performance decreases as the number of episodes increases, further indicating stable learning outcomes.

2) Effectiveness of the proposed solution: To demonstrate the effectiveness of the proposed solution, we conducted simulations with different settings for the UAV's bandwidth budget and the required CPU cycles of the tasks. The bar chart in Fig. 3 illustrates the superior performance of the proposed solution in maximising reliable task offloading portions under various bandwidth allocation schemes and CPU requirements for computational tasks, compared to the benchmark scheme. The comparison is made between Max $C_i = 900$ megacycles and Max $C_i = 1200$ megacycles. The results demonstrate

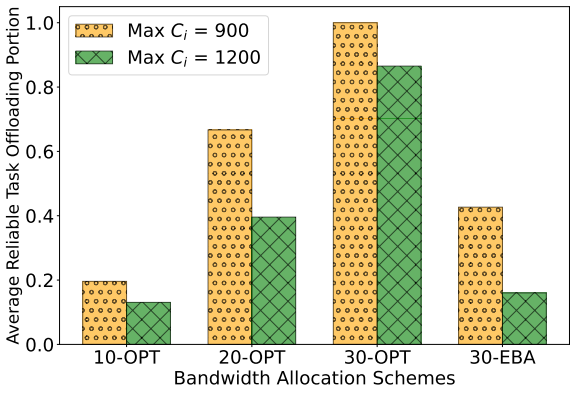

Fig. 3. The effectiveness of the proposed solution in maximising the reliable task offloading portions with different settings of $B_j^{\max}$ and maximum CPU required by the computational tasks in the scenario of $N = 15$ GUs. Here, "10-OPT" represents the optimal bandwidth allocation scheme with $B_j^{\max} = 10$ MHz and "30-EBA" represents the equal bandwidth allocation scheme with $B_j^{\max} = 30$ MHz.

that the optimal allocation schemes outperform the equal allocation strategy. For instance, the 30-OPT scheme achieves the highest reliable offloading portion, reaching nearly 1.0 for Max $C_i = 900$, while 30-EBA shows significantly lower performance in both cases. This highlights the efficiency of the proposed solution in utilising resources to improve reliable task offloading. In addition, Fig. 3 demonstrates how the UAV's bandwidth budget affects the offloading process. As shown in the figure, increasing the bandwidth budget for GU-to-UAV communication significantly enhances the reliability of task offloading, allowing a higher portion of tasks to be fully offloaded to complete the task within the delay tolerance.

V. Conclusion

In conclusion, we have investigated the optimal design of task offloading and bandwidth allocation for reliability-driven SAGIN-enabled edge computing. The proposed system model takes into account the dynamic environment of computing demands, the energy budgets of GUs, and the computing capacity of the satellite's edge server. Our DRL-based solution provides an optimal approach to optimising offloaded task portions and bandwidth allocations, thereby enhancing the reliability of the offloading process within the system. The effectiveness of the proposed solution has been clearly demonstrated through simulation results, which show stable training patterns and maximised reliability in task offloading. Lastly, we have shown that, the development of real-time optimisation for UAV deployment presents a promising future direction to further enhance system efficiency and adaptability in dynamic environments.

References

[1] Y. Liu, L. Jiang, Q. Qi, K. Xie, and S. Xie, "Online computation offloading for collaborative space/aerial-aided edge computing toward 6G system," *IEEE Trans. Veh. Technol.*, vol. 73, no. 2, pp. 2495–2505, Feb. 2024.

[2] T. Ma, H. Zhou, B. Qian, N. Cheng, X. Shen, X. Chen, and B. Bai, "UAV-LEO integrated backbone: A ubiquitous data collection approach for B5G Internet of remote things networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3491–3505, Nov. 2021.

[3] Q. Chen, Z. Guo, W. Meng, S. Han, C. Li, and T. Q. S. Quek, "A survey on resource management in joint communication and computing-embedded SAGIN," *IEEE Commun. Surveys Tuts.*, 2024.

[4] T. Do-Duy, D. V. Huynh, E. Garcia-Palacios, T.-V. Cao, V. Sharma, and T. Q. Duong, "Joint computation and communication resource allocation for unmanned aerial vehicle NOMA systems," in *Proc. IEEE 28th Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD)*, Edinburgh, United Kingdom, Nov. 2023, pp. 290–295.

[5] B. Shang, Y. Yi, and L. Liu, "Computing over space-air-ground integrated networks: Challenges and opportunities," *IEEE Netw.*, vol. 35, no. 4, pp. 302–309, Aug. 2021.

[6] J. He, N. Cheng, Z. Yin, C. Zhou, H. Zhou, W. Quan, and X.-H. Lin, "Service-oriented network resource orchestration in space-air-ground integrated network," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 1162–1174, Jan. 2024.

[7] F. Tang, H. Hofner, N. Kato, K. Kaneko, Y. Yamashita, and M. Hangai, "A deep reinforcement learning-based dynamic traffic offloading in space-air-ground integrated networks (SAGIN)," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 276–289, Jan. 2022.

[8] B. Cao, J. Zhang, X. Liu, Z. Sun, W. Cao, R. M. Nowak, and Z. Lv, "Edge–cloud resource scheduling in space–air–ground-integrated networks for internet of vehicles," *IEEE Internet of Things J.*, vol. 9, no. 8, pp. 5765–5772, Apr. 2022.

[9] C. Huang, G. Chen, P. Xiao, Y. Xiao, Z. Han, and J. A. Chambers, "Joint offloading and resource allocation for hybrid cloud and edge computing in SAGINs: A decision assisted hybrid action space deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 5, pp. 1029–1043, May 2024.

[10] I. Leyva-Mayorga, M. Martinez-Gost, M. Moretti, A. Pérez-Neira, M. Ángel Vázquez, P. Popovski, and B. Soret, "Satellite edge computing for real-time and very-high resolution earth observation," *IEEE Trans. Commun.*, vol. 71, no. 10, pp. 6180–6194, Oct. 2023.

[11] Y. Gao, Z. Yan, K. Zhao, T. de Cola, and W. Li, "Joint optimization of server and service selection in satellite-terrestrial integrated edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 2, pp. 2740–2754, Feb. 2024.

[12] T. Q. Duong, L. D. Nguyen, T. T. Bui, K. D. Pham, and G. K. Karagiannidis, "Machine learning-aided real-time optimized multibeam for 6G integrated satellite-terrestrial networks: Global coverage for mobile services," *IEEE Netw.*, vol. 37, no. 2, pp. 86–93, Apr. 2023.

[13] Z. Song, Y. Hao, Y. Liu, and X. Sun, "Energy-efficient multiaccess edge computing for terrestrial-satellite internet of things," *IEEE Internet of Things J.*, vol. 8, no. 18, pp. 14 202–14 218, Sep. 2021.

[14] C. Ding, J.-B. Wang, H. Zhang, M. Lin, and G. Y. Li, "Joint optimization of transmission and computation resources for satellite and high altitude platform assisted edge computing," *IEEE Trans. Commun.*, vol. 21, no. 2, pp. 1362–1377, Feb. 2022.

[15] F. Chai, Q. Zhang, H. Yao, X. Xin, R. Gao, and M. Guizani, "Joint multi-task offloading and resource allocation for mobile edge computing systems in satellite IoT," *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 7783–7795, Jun. 2023.

[16] P. Zhang, N. Chen, S. Shen, S. Yu, N. Kumar, and C.-H. Hsu, "Ai-enabled space-air-ground integrated networks: Management and optimization," *IEEE Netw.*, vol. 38, no. 2, pp. 186–192, Apr. 2024.

[17] M.-H. T. Nguyen, T. T. Bui, L. D. Nguyen, E. Garcia-Palacios, H.-J. Zepernick, H. Shin, and T. Q. Duong, "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.

[18] D. V. Huynh, V.-D. Nguyen, S. Chatzinotas, S. R. Khosravirad, H. V. Poor, and T. Q. Duong, "Joint communication and computation offloading for ultra-reliable and low-latency with multi-tier computing," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 521–537, Feb. 2022.

[19] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, Oct. 2018.

[20] T. Q. Duong, D. V. Huynh, Y. Li, E. Garcia-Palacios, and K. Sun, "Digital twin-enabled 6G aerial edge computing with ultra-reliable and low-latency communications," in *Proc. 2022 1st International Conference on 6G Networking (6GNet)*, Paris, France, Jul. 2022.