Task Offloading and Resource Allocation for Space-Air-Ground Integrated Network

Boyu Liu

A thesis submitted in fulfilment of the requirements for the degree of Master of Philosophy

Supervisors:
Dr. Yonghui Li
Prof. Branka Vucetic

Faculty of Engineering
School of Electrical and Computer Engineering
The University of Sydney

2025

To my parents and grandparents.

Statement of Originality

To the best of my knowledge, this thesis does not include materials that have been submitted for any degree or other purposes except where due acknowledgment has been made in the text.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Boyu Liu

Signature:

As a supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Yonghui Li

Signature:

1 July 2025

Authorship Attribution Statement

I, Boyu Liu, declare that the contents of this thesis are my own work. I am the lead author for the conference papers covered in this thesis as a Master of Philosophy (MPhil) candidate in the School of Electrical and Computer Engineering at the University of Sydney. The main contributions of this thesis are presented in Chapters 3 and 4.

• Chapter 3 presents a dynamic graph-based reinforcement learning in SAGIN resource allocation. This work is presented in [C1].

• Chapter 4 develops a rule- and threshold-driven *Dynamic Functional Switching Mechanism* (DFSM) that enables role-aware operation (computation/relay/caching) and integrates it with DHGNN for role-consistent decision making in SAGIN.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Boyu Liu

Signature:

1 July 2025

As a supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Yonghui Li

Signature:

1 July 2025

Abstract

Space-air-ground integrated networks (SAGIN) have emerged as a key enabling technology for 6G communications, providing seamless global connectivity through the integration of space-based satellites, airborne platforms, and terrestrial networks. SAGIN holds significant promise for diverse applications including disaster recovery, military communications, Internet of Things (IoT), and extended reality (XR) services. Despite these advantages, SAGIN faces substantial operational challenges due to its inherent heterogeneity, dynamically changing network topology, and high node mobility, all of which complicate efficient task offloading and resource allocation (TORA). These challenges are exacerbated by stringent quality-of-service (QoS) demands, limited spectrum resources, and constrained power availability, making effective resource management particularly challenging.

In this thesis, we study the state of the art in resource management across 5G and SAGIN, with a focused literature review that spans core/RAN resource allocation in 5G and task offloading/resource allocation (TORA) in SAGIN using optimization-based, deep reinforcement learning (DRL)-based, and GNN-augmented DRL approaches. We find that existing methods often rely on static or homogeneous abstractions of SAGIN, under model temporal topology changes, do not encode node-type heterogeneity explicitly, and lack mechanisms to enforce role-consistent feasibility; furthermore, many methods suffer from large action spaces

and high inference complexity that hinder real-time deployment under latency and energy constraints.

We formulate a comprehensive multi-objective optimization framework addressing the complex TORA problem within SAGIN, focusing specifically on minimizing communication latency and inter-node power consumption. We introduce a novel dynamic heterogeneous graph neural network (DHGNN)-based reinforcement learning approach. Our DHGNN dynamically processes real-time agent states, systematically updates node features, and explicitly captures complex dependencies between diverse node types (User Equipment, High-Altitude Platforms, and Low-Earth Orbit satellites) through tailored heterogeneous graph representations. Unlike traditional static or homogeneous graph-based methods, our DHGNN approach effectively encodes both temporal network dynamics and structural heterogeneity, significantly improving the accuracy and efficiency of state representations and predictions.

We integrate the DHGNN architecture into a reinforcement learning framework, enabling adaptive, real-time decision-making that optimizes the competing objectives of latency minimization and power consumption reduction. To facilitate this optimization, we propose a periodic topology update mechanism explicitly designed to handle node mobility efficiently, balancing computational complexity with performance gains. The framework continuously adapts to dynamic changes in channel conditions, resource availability, and network topology, ensuring robust and scalable resource allocation decisions.

We further propose a Dynamic Functional Switching Mechanism (DFSM) that enforces role-consistent feasibility and enhances runtime efficiency. DFSM assigns each non-UE node an exclusive role among *computation*, *relay*, and *caching*, and updates the roles using rules and threshold-driven logic with window updates, cooldown, and hysteresis. The resulting role masks constrain the feasible action

set and are exposed to the policy state, improving robustness under time-varying operating conditions.

Extensive simulations demonstrate that the proposed DHGNN framework substantially outperforms existing graph deep reinforcement learning (GDRL) baselines in cumulative reward, convergence speed, and stability, while reducing average latency and power consumption. When the DFSM enabled, additional gains are observed in dynamic settings (higher final rewards in both dynamic heterogeneous and homogeneous graphs) together with consistent runtime benefits, reducing inference latency by 26–33% and power consumption by 24–42% across configurations. These results highlight the effectiveness and practicality of jointly modeling SAGIN's temporal dynamics and node-type heterogeneity and enforcing role-consistent feasibility, yielding a robust and scalable solution for future 6G SAGIN deployments.

Acknowledgements

First and foremost, I want to thank Dr. Yonghui Li for his consistent encouragement and guidance throughout my research journey. His expertise and mentorship have provided me with the necessary understanding of proper research practices. I am grateful for his numerous ideas and assistance in paper writing. Likewise, I sincerely thank Prof. Branka Vucetic for her professional advice and insightful suggestions.

I would also like to acknowledge the support and contributions of my academic advisor Peng Cheng. During my MPhil candidature, Peng provided valuable suggestions that have significantly enhanced my research.

I am grateful to The University of Sydney for allowing me to study at one of the top universities in the world.

Finally, I express my deepest gratitude to my parents and grandparents for allowing me to study abroad and achieve certain milestones.

List of Publications

Conference Papers

The paper was accepted by IEEE VTC.

B. Liu, P. Cheng, W. Xiang, B. Vucetic, and Y. Li, "Dynamic Heterogeneous Graph Learning for Multi-object Resource Allocation in Space-Air-Ground-Integrated Networks," accepted to appear in Proc. IEEE Vehicular Technology Conference (VTC2025-Fall), Chengdu, China, 2025.

As the first author, I was primarily responsible for the majority of the work presented in the aforementioned publications. My contributions encompassed the conception and design of the research, development of the theoretical framework and mathematical derivations, implementation of software and simulations, interpretation of results, and preparation of the initial manuscript drafts.

Signature:

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Signature:

Contents

List of Abbreviations

1G First Generation (analog cellular)

2G Second Generation (digital cellular)

3G Third Generation

4G Fourth Generation

5G Fifth Generation

6G Sixth Generation

AWGN Additive White Gaussian Noise

BPv7 Bundle Protocol Version 7

CFO Carrier Frequency Offset

CMDP Constrained Markov Decision Process

CMOS Complementary Metal–Oxide–Semiconductor

CPU Central Processing Unit

CSI Channel State Information

DFSM Dynamic Node Functional Switching Mechanism

DHGNN Dynamic Heterogeneous Graph Neural Network

DHOM Dynamic Homogeneous Graph Model

DQN Deep Q-Network

DRL Deep Reinforcement Learning

DTN Delay-/Disruption-Tolerant Networking

GDRL Graph Deep Reinforcement Learning

GEO Geostationary Earth Orbit

GNN Graph Neural Network

HAPS High-Altitude Platform Station

ISL Inter-Satellite Link

LEO Low Earth Orbit

LoS Line-of-Sight

LTE Long-Term Evolution

MDP Markov Decision Process

MAC Medium Access Control

MEC Multi-access Edge Computing

MLP Multi-Layer Perceptron

NR New Radio

NTN Non-Terrestrial Network

OFDM Orthogonal Frequency-Division Multiplexing

PAT Pointing, Acquisition, and Tracking

PSO Particle Swarm Optimization

QoS Quality of Service

RAN Radio Access Network

RL Reinforcement Learning

SAGIN Space–Air–Ground Integrated Network

SHEM Static Heterogeneous Graph Model

SHOM Static Homogeneous Graph Model

ST-GCN Spatio-Temporal Graph Convolutional Network

UAV Unmanned Aerial Vehicle

UE User Equipment

UMTS Universal Mobile Telecommunications System

UPA Uniform Planar Array

XR Extended Reality

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Evolution from 1G to 5G

Cellular systems have advanced through successive generations, each eliminating key bottlenecks and expanding service capabilities. The first generation (1G) delivered analog voice; 2G digitized access and introduced text messaging; 3G brought packet data (UMTS/cdma2000) and mobile multimedia; 4G completed the all-IP transition with LTE, enabling broadband mobility and flexible interworking among heterogeneous access networks. Against this historical backdrop, 5G did not merely raise throughput: it formalized three distinct service families—enhanced mobile broadband (eMBB), ultra-reliable low-latency communications (URLLC), and massive machine-type communications (mMTC)—and introduced the architectural capabilities for programmability and softwarization across the radio and core, including network slicing and edge computing. Of particular relevance to global service continuity, the 5G standardization stream initiated explicit treatment of non-terrestrial networks (NTN), acknowledging satellite and high-altitude platform integration and documenting their distinctive channel traits

such as long propagation delays and Doppler dynamics [1]–[4]. These moves built on the maturation of IP convergence and SDN/NFV control, as well as incremental evidence from decades of satellite–cellular interworking research that multi-domain connectivity could be engineered within a unified protocol fabric [1], [2], [4]. With these foundations established, we next examine the limitations of current deployments and their implications for a more integrated architecture.

1.2 Limitations of 5G Networks

Despite these advances, 5G retains structural constraints that compromise universal service and motivate a more integrated architecture. Coverage and resilience gaps persist in rural, maritime, polar, and disaster-stricken regions where terrestrial footprints are sparse or fragile; even with NTN features, de facto deployments remain dominated by ground infrastructure , and restoration after large-scale failures can be lengthy [1], [3]. End-to-end latency guarantees are uneven across domains: while terrestrial paths can meet sub–10 ms budgets for URLLC, routes that traverse long-haul backhaul or high-delay segments risk violating strict deadlines unless traffic is steered intelligently and computation is offloaded to the edge [1], [2]. Interoperability across space, air, and ground imposes nontrivial control-plane overhead due to disparate delay, Doppler, and reliability profiles, and mobility management must contend with visibility windows and frequent handovers that are qualitatively different from terrestrial patterns [2], [3]. Finally, joint orchestration of spectrum, power, and computation at 5G scale remains combinatorial and non-stationary in dense multi-segment deployments, stressing classical optimization pipelines designed for more static and homogeneous settings [5], [6]. These structural constraints indicate that merely scaling 5G is insufficient; instead, a comprehensive cross-domain integration is needed—a direction central to emerging 6G visions and concretely embodied by SAGIN.

1.3 6G and Space-Air-Ground Integrated Network

The vision for 6G shifts from incremental KPI improvements to integration: a unified, intelligent architecture that combines the complementary strengths of space, air, and ground. Within this vision, the Space–Air–Ground Integrated Network (SAGIN) provides the architectural framework that couples global reach (satellites and HAPS), metropolitan capacity (advanced terrestrial radio and fiber), and adaptability (software-defined, AI-driven control) into a single service plane [6]–[8]. Low-Earth-Orbit (LEO) constellations such as Starlink and OneWeb further this trajectory by creating low-latency space meshes with optical inter-satellite links, enabling in-space routing, direct user access in selected markets, and resilient backhaul augmentation. In certain geodesic routes, direct free-space paths can yield end-to-end delays comparable to or shorter than long terrestrial detours, highlighting the performance complementarity of the space layer when co-optimized with ground infrastructure [9], [10]. As 3GPP NTN work continues to codify interfaces and procedures, these constellations become first-class components in a multi-domain 6G ecosystem [3]. To ground this vision, we next revisit the lineage of ideas and prototypes that foreshadowed space–air–ground integration and informed today's standardization efforts.

1.3.1 Space-Air-Ground Integrated Network History

The concept of SAGIN predates current 6G discussions. In the 1990s and early 2000s, Satellite-UMTS proposals and inter-segment handover algorithms explored ways to integrate satellite and cellular systems while coping with delay and mobility asymmetries [11]. High-altitude platform stations (HAPS) were investigated as quasi-stationary aerial relays offering broad coverage and favorable line-of-sight

conditions at lower latency than GEO links, while providing greater stability than low-flying UAVs, thereby expanding the design space for non-terrestrial augmentation [12]. With the 4G all-IP core, prototype hybrids such as satellite–LTE demonstrated the feasibility of common protocol stacks and shared control planes across domains [13]. The late 2010s saw these strands consolidated into formal SAGIN architectures and comprehensive surveys, reframing space (LEO/MEO/GEO), air (UAVs/HAPS), and ground segments as a coordinated system rather than ad hoc adjuncts [4], [7]. In parallel, standardization documents began embedding NTN into the cellular framework and to recognize the specific engineering mechanisms required for interoperable services, including waveform adaptations, timing and frequency tracking under high Doppler, and mobility procedures for intermittent visibility [1], [3]. *With this architectural foundation in place, the next focus is on the central challenge: how to enable effective task offloading and resource allocation in operational SAGINs.*

1.3.2 Space-Air-Ground Integrated Network Overview

The Space-Air-Ground Integrated Network (SAGIN) is an innovative wireless networking framework designed to overcome coverage, capacity, and resilience challenges associated with terrestrial-only communication systems. SAGIN incorporates three hierarchical network layers: a space segment (satellite communication systems), an air segment (high-altitude platform stations, HAPS, and unmanned aerial vehicles, UAVs), and a terrestrial segment (traditional ground-based cellular networks) [7]. By seamlessly integrating these layers, SAGIN harnesses the complementary strengths of each to deliver seamless global connectivity, high reliability, and superior flexibility compared to standalone terrestrial networks.

Each layer within SAGIN offers distinct capabilities:

1. Space Segment: Satellites, particularly low Earth orbit (LEO) constellations, provide extensive geographical coverage, enabling connectivity even in remote and sparsely populated regions. Satellites offer resilient links during large-scale disasters when terrestrial networks are disrupted [7].

2. Air Segment: Airborne platforms, including HAPS and UAVs, serve as flexible intermediate nodes, rapidly deploying temporary communication links. Their mobility and line-of-sight communication characteristics are essential to cover dynamic hotspots, disaster areas, and regions with temporarily high traffic demands [14].

3. Ground Segment: Terrestrial networks, particularly cellular networks, provide high-capacity connectivity, low latency, and superior quality of service (QoS) for densely populated urban environments, supporting data-intensive applications.

The integration of these heterogeneous network layers enables SAGIN to meet diverse service requirements, from latency-critical real-time communication to massive data transfers and emergency services [15]. Potential application scenarios include rural connectivity, maritime and aerial communications, emergency response, environmental monitoring, intelligent transportation systems, and smart city initiatives.

1.4 Space-Air-Ground Integrated Network Task Offloading and Resource Allocation

Within the integrated architecture, task offloading and resource allocation (TORA) is a key mechanism for transforming heterogeneous connectivity into high-quality end-to-end services. Effective TORA must accommodate a connectivity graph that changes on orbital and aerial timescales, with link budgets and feasible paths

reshaped by LEO motion and aerial maneuvering; decisions that were valid minutes earlier may become inefficient or invalid as visibility windows shift and handovers cascade. It must reconcile cross-domain heterogeneity, since links differ in latency, bandwidth, and reliability, while nodes differ in onboard compute, energy, and caching capability—ranging from bent-pipe payloads to regenerative processing. Policies therefore need to be role-aware and context-sensitive to prevent inefficient placements and bottlenecks [2], [8], [12]. The problem is intrinsically multi-objective: minimizing latency and energy while meeting QoS targets across domains yields a large, time-varying, and often non-convex decision space.

1.5 Research Problem and Gaps

SAGIN is considered a promising solution to address the limitations of traditional terrestrial wireless networks in terms of global coverage, service continuity, and resilience **to** failures. By integrating satellites, aerial platforms (such as HAPS and UAVs), and terrestrial infrastructure, SAGIN enables integrated communication across diverse environments. However, to fully realize its potential, SAGIN still faces several critical and unresolved research problems.

A primary challenge lies in accurately modeling the dynamic and heterogeneous nature of SAGIN. Existing network representations often assume static or homogeneous structures, failing to reflect the distinct characteristics and behavior across satellite, aerial, and ground nodes. These models oversimplify the underlying system and overlook the structural diversity of SAGIN components, such as differences in mobility patterns, energy constraints, computational capabilities, and communication roles. As a result, current models are inadequate for capturing the full complexity of real-world SAGIN deployments, limiting their effectiveness in guiding system-level decision-making and optimization.

Moreover, the roles and capabilities of SAGIN nodes are dynamic—they may vary over time depending on network conditions, task demands, and operational constraints. For instance, an aerial platform may serve as a communication relay under normal conditions but act as a computing or caching node when necessary. Despite this inherent functional flexibility, most existing methods adopt static role assignments for each type of node, without accounting for role adaptation to environmental dynamics. The lack of mechanisms to support flexible function switching impairs the network's ability to respond efficiently to unpredictable events such as sudden traffic surges or node failures.

Another significant issue arises from the highly dynamic SAGIN topology. Due to the mobility of satellites and aerial platforms, the network experiences frequent changes in link availability, path stability, and node connectivity. These dynamics complicate task scheduling and resource allocation decisions, which are often based on static or slowly varying assumptions in traditional approaches. Ensuring stable performance under rapid topological changes remains an open challenge, particularly for low-latency, high-reliability scenarios.

Finally, making effective real-time decisions under such dynamic and heterogeneous conditions remains highly challenging. Conventional optimization-based solutions often require complete and static information and are computationally intractable for large-scale, time-sensitive scenarios. In practice, decisions on task offloading, route selection, and function switching must be made quickly and continuously, under uncertainty and partial observability. This introduces a gap between the theoretical formulations of resource allocation and their feasibility in operational SAGIN systems.

These challenges—including the lack of fine-grained network modeling, the inability to adapt node functionality, the instability introduced by dynamic topologies, and the need for real-time scalable decision-making—highlight an urgent need in

current SAGIN research. It is essential to develop an integrated framework that can accurately represent heterogeneous network components, dynamically adapt to role changes, and support intelligent decision-making in evolving and complex environments.

1.6 Contributions

To address the key challenges identified in the previous section—namely the lack of fine-grained heterogeneous modeling, the inability to adapt node functionality, and the difficulty of making real-time decisions in dynamic network topologies—this dissertation proposes an integrated research framework composed of novel modeling techniques, adaptive decision mechanisms, and learning-based resource orchestration. The major contributions of this work are as follows:

1. A dynamic heterogeneous graph framework for SAGIN modeling. This dissertation presents a novel dynamic heterogeneous graph model that captures the structural and behavioral and functional complexity of SAGIN. In contrast to conventional network models that treat nodes uniformly or statically, the proposed framework explicitly represents three distinct types of nodes—LEO satellites, HAPS/UAVs, and terrestrial user equipment (UEs)—as heterogeneous entities with differing capabilities, roles, and mobility patterns. The dynamic graph formulation aexplicitly models temporal variations in connectivity, task generation, and resource availability, thereby enabling a more accurate and adaptive system representation. This modeling approach lays the groundwork for scalable algorithms that better reflect real-world SAGIN conditions and provides a unified representation across multiple abstraction layers.

2. A dynamic node function conversion mechanism for role-aware adaptation. To support flexible and context-aware operations, this work proposes a

mechanism that enables nodes to dynamically switch between communication, computation, and caching roles based on local observations and global objectives. Instead of assuming fixed functionality for satellites or aerial nodes, the system evaluates each node's current state—such as channel quality, energy budget, and proximity to other nodes—and determines the most beneficial role configuration. This mechanism significantly enhances resource utilization and enables the network to respond to changing environments, such as network congestion, link failures, or sudden task surges. The decision logic for function conversion is integrated into the learning and optimization pipeline, rather than treated as a heuristic.

3. A reinforcement learning framework for joint task offloading and resource allocation. To address the complexity and uncertainty of real-time decision-making in dynamic environments, a deep reinforcement learning (DRL) framework is developed to jointly optimize task offloading and multi-resource resource allocation (spectrum, energy, compute). The DRL agent learns policies through continuous interaction with the environment, guided by a reward function that balances task completion delay, energy consumption, and system throughput. Unlike rule-based or offline optimization methods, the learning-based approach can adapt to previously unseen network conditions, learn temporal patterns of network dynamics, and handle stochastic task arrivals and link fluctuations. The model supports both centralized training and distributed inference settings, offering scalability for large-scale deployments.

4. An end-to-end system implementation and comprehensive experimental validation. All proposed components—including the heterogeneous graph representation, the function-switching mechanism, and the learning-based decision engine—are integrated into a cohesive end-to-end framework. The system is implemented and evaluated through extensive simulation studies under realistic SAGIN settings, including mobile satellite constellations, varying user densities, and mixed traffic profiles (latency-sensitive vs. throughput-driven tasks). Experi-

mental results show that the proposed framework significantly outperforms several baseline methods in terms of task success rate, average delay, energy efficiency, and adaptability to network changes. Ablation studies further validate the importance of each module, and sensitivity analysis confirms the robustness of the proposed solutions under diverse operating conditions.

Collectively, these contributions advance the modeling fidelity, decision intelligence, and practical feasibility of SAGIN systems. By combining structural network representations with learning-based control policies, this work paves the way for more autonomous, scalable, and efficient future communication networks that operate reliably across space, air, and ground domains.

Mapping Between Challenges and Contributions

The four contributions of this dissertation directly address the research gaps identified in Section 1.5. First, the proposed **dynamic heterogeneous graph framework** (Contribution 1) resolves **Challenge 1**, which concerns the lack of fine-grained heterogeneous modeling. By explicitly distinguishing LEO, HAPS/UAV, and UE nodes and capturing time-varying connectivity, it provides the structural fidelity required for realistic SAGIN decision-making.

Second, the **dynamic node function switching mechanism** (Contribution 2) addresses **Challenge 2**, namely the inability of existing approaches to adapt node roles over time. By enabling nodes to switch among computation, relay, and caching functions based on real-time conditions, this mechanism enhances flexibility and improves resource utilization under fluctuating workloads and channel quality.

Third, the **deep reinforcement learning framework** (Contribution 3) jointly tackles **Challenge 3** and **Challenge 4**. Dynamic topology variations are miti-

gated through RL's ability to learn adaptive policies that respond to rapid link-state changes, while the need for real-time scalable decision-making is met through fast inference and online adaptability. Unlike static optimization, the learning-based approach remains robust under uncertainty and partial observability.

Finally, the **end-to-end implementation and comprehensive validation** (Contribution 4) reinforce all previous contributions by demonstrating their practicality, supporting scalability, and confirming that the proposed solutions remain effective across diverse dynamic SAGIN scenarios. This integrated evaluation closes the loop between modeling, algorithm design, and system-level feasibility.

Together, these contributions provide a coherent response to the four identified challenges and establish a unified foundation for intelligent, flexible, and scalable resource management in realistic SAGIN environments.

1.7 Thesis Outline

Chapter 1 explains the motivation for research and research problems. A brief overview of history is also presented.

Chapter 2 investigates the literature review of SAGIN.

Chapter 3 presents a dynamic heterogeneous graph learning mechanism. We analyze the convergence reward from reinforcement learning and compare the performance of the proposed algorithm with existing algorithms.

Chapter 4 discusses the other contribution of my thesis, which is dynamic node function switch mechanism, including motivation, system model, and results.

Chapter 5 summarizes the thesis and provides some insight into future work.

Chapter 2

Literature Review

In the previous chapters, we introduced the development of SAGIN and discussed its significance for future 6G systems. We emphasized the seamless integration of satellite, aerial, and terrestrial networks as a key enabler for ubiquitous connectivity and resilient services. At the same time, the dynamic and heterogeneous nature of SAGIN raises unique challenges—most prominently in task offloading and resource allocation—which makes adaptive and efficient orchestration indispensable.

Building on this motivation, this chapter provides a comprehensive literature review focused on resource allocation for both 5G networks and SAGIN, and—new in this version—adds two concise reviews of enabling directions that directly affect the feasibility and performance of resource management in practice. First, we survey resource allocation strategies in 5G, covering both the core network and the radio access network (RAN); We then examine task offloading and resource allocation within the SAGIN framework. Following the thesis focus, we structure this part into three subsections: *Traditional Optimization Approaches in SAGIN*, *Deep Reinforcement Learning for SAGIN*, and *Graph Neural Network-Augmented*

Deep Reinforcement Learning in SAGIN. A table at the beginning of that section-summarizes problem settings and techniques, and an additional table at the end highlights distinctive properties of learning-based approaches.

2.1 NTN Channel Access and Modeling

2.1.1 PHY/MAC Adaptations for NR–NTN

Integrating low Earth orbit (LEO) satellite links into New Radio (NR) necessitates adapting access procedures to accommodate long propagation delays and large Doppler shifts. 3GPP has introduced NTN-oriented enhancements, including extended timing advance, relaxed HARQ timing, global navigation satellite system/ephemeris-aided frequency pre-compensation, and adapted random access (RA) proceduresto ensure link reliability and robust access under high mobility and varying slant ranges. These adaptations enable bent-pipe or regenerative payloads to interoperate with terrestrial NR stacks while maintaining service continuity across space/air/ground segments [3], [16].

2.1.2 Waveform, Synchronization, and Doppler Compensation

LEO dynamics induce frequency offsets that exceed conventional OFDM tolerances if left uncompensated. Standardized solutions combine network-assisted carrier frequency advance, differential Doppler compensation across wide beams, and GNSS-informed pre-correction at the UE to keep residual CFO/ICI within demodulation limits, while relaxing tracking loop requirements and reducing excessive control overhead. These mechanisms sustain synchronization, scheduling,

and RA performance in mixed satellite/terrestrial deployments, enabling NTN cells to behave as NR entities in SAGIN scenarios [3], [16].

2.1.3 Channel and Link Modeling Across GEO/LEO/HAPS

Accurate channel abstractions support numerology design, link adaptation, and system-level evaluation in SAGIN. 3GPP studies extend terrestrial models by accounting for long delay spreads, LoS-dominant Rician fading with elevation-dependent path loss, blockage statistics, platform mobility, and beamfootprint dynamics. Such models support feasibility analyses for coverage, HARQ viability, and multi-connectivity across space/air/ground layers [3], [16].

2.2 Inter-Satellite Networking and Routing in LEO Constellations

2.2.1 Snapshot- and Time-Expanded Routing

LEO constellations with inter-satellite links (ISLs) form time-varying dynamic space networks. Classical designs partition orbital motion into quasi-static "snapshots", precomputing shortest paths for each; recent approaches generalize this to time-expanded graphs that encode space–time edges and schedule predictable multi-hop routes across orbital dynamics [17], [18]. Predictive routing reduces control overhead and avoids path oscillations while keeping end-to-end latency competitive with terrestrial backhaul in selected corridors [18].

2.2.2 Delay/Disruption-Tolerant Networking (DTN)

Because ISL and gateway visibility can be intermittent, DTN provides a store–carry–forward overlay, tolerant of disconnections and long round-trip times. The Bundle Protocol v7 (BPv7) defines custody transfer, timers, and convergence layers that bridge heterogeneous links and long delays; it has been investigated as a session layer for LEO backbones that feed terrestrial domains in SAGIN [18], [19]. DTN complements predictive routing by ensuring robust delivery when partial connectivity or link outages occur.

2.2.3 ISL Topology Control and Optical Link Scheduling

Laser ISLs offer high capacity but require precise pointing, acquisition, and tracking (PAT), motivating topology-control strategies that selectively activate ISLs to balance coverage, latency, and robustness. Recent surveys synthesize techniques for adaptive ISL activation, congestion-aware path selection, and gateway integration, describing a 'satellite Internet architecture', interoperating with 3GPP / IETF stacks[18], [20]. These space-segment advances create a global relay fabric that complements terrestrial and air-to-ground links within SAGIN.

2.3 Task Offloading and Resource Allocation in SAGIN

2.3.1 Traditional Optimization Approaches of TORA in SAGIN

Space–Air–Ground Integrated Networks (SAGIN) combine satellite, aerial and terrestrial networks into a unified architecture to provide ubiquitous coverage and computing capabilities [21]. Managing computational offloading and resource allocation in SAGIN is challenging due to heterogeneous nodes (LEO satellites, high-altitude platforms, UAVs, and ground base stations) and dynamic network conditions.

Early works formulated these problems as mathematical optimization models, leveraging convex optimization techniques to obtain optimal or near-optimal solutions under certain assumptions. For example, resource allocation for integrated satellite-aerial-ground networks has been cast as a convex program, maximizing connectivity and throughput[22]. In such formulations, relaxations and decomposition methods (e.g., Lagrangian duality or successive convex approximation) are used to handle coupling between communication and computing variables, yielding efficient solutions when the problem structure is tractable.

However, many SAGIN optimization tasks, such as joint user association, task offloading decision, and multi-tier resource scheduling, are NP-hard combinatorial problems. Traditional exact methods (e.g., brute-force search or mixed-integer programming) become infeasible at scale due to the exponential solution space size. Hence, researchers have turned to heuristic and metaheuristic algorithms to find high-quality approximate solutions within reasonable timeframes. In particular, population-based search algorithms such as Genetic Algorithms (GA),

Simulated Annealing (SA), and Particle Swarm Optimization (PSO) have been widely adopted for SAGIN resource management problems [23]–[25]. These approaches iteratively improve candidate offloading schedules or resource allocation plans and can effectively navigate the large solution space induced by multiple offloading options across network layers. For instance, Du *et al.* develop an enhanced GA that co-optimizes task assignment and transmission scheduling across a multi-layer (space–air–ground–sea) network, achieving significant improvements in secure offloading performance compared to baseline SA and PSO methods [25]. Likewise, PSO-based heuristics have been applied to jointly schedule tasks and position UAV aerial servers in SAGIN, demonstrating near-optimal latency and load-balancing with much lower complexity than exhaustive search [23]. In addition, SA algorithms have been explored for specific sub-problems such as routing optimization in LEO satellite–assisted UAV networks, where SA efficiently finds routes that minimize end-to-end delay in a highly dynamic SAGIN environment [24].

These traditional optimization approaches have been validated in various real-world SAGIN application scenarios. In satellite-dominated deployments (e.g., LEO satellite constellations providing edge computing at the network edge), convex optimization and greedy scheduling are used to allocate channel and computing resources across space and ground segments, improving global coverage and reducing service delay [22]. In UAV-assisted SAGIN networks, where drones act as airborne MEC servers, researchers have shown that heuristic offloading strategies can adapt to time-varying user demands and flight constraints, effectively splitting workloads between aerial and terrestrial nodes [26]. For example, Fan *et al.* propose a UAV-assisted traffic offloading scheme, dynamically assigning users' tasks to UAVs or ground base stations, achieving lower latency for delay-sensitive traffic through intelligent resource allocation [26]. Moreover, in emergency and disaster-recovery scenarios, classical optimization frameworks have been employed to max-

imize network throughput and task completion under infrastructure damage and surging demand. Zhang *et al.* [27] consider a post-disaster SAGIN environment (with LEO satellites and high-altitude platforms supporting ground users) and formulate a joint user association and offloading decision problem. By solving this with a tailored optimization algorithm, their approach ensures critical tasks are offloaded to aerial or space nodes when terrestrial links are impaired, significantly improving service resilience during emergencies.

Cheng et al. [6] conducted a comprehensive review of these techniques, identifying their strengths in theoretical tractability but also noting the high computational overhead and poor adaptability to dynamic environments. The limitation is particularly pronounced in scenarios where node mobility, variable link quality, and intermittent connectivity render static optimization solutions ineffective. In real-world applications, such as UAV-based emergency communications or LEO satellite relaying, centralized solvers often cannot react quickly enough to topological changes or user mobility. Moreover, these approaches typically assume perfect channel state information (CSI) and global network knowledge, assumptions rarely valid in practice.

In summary, conventional optimization techniques—ranging from rigorous convex optimizers, to bio-inspired metaheuristics—play a crucial role in current SAGIN research. They provide systematic means to handle the complex decision-making in task offloading and resource allocation across the space, air, and ground tiers. These methods have achieved notable success in improving performance metrics (such as latency, energy consumption, and throughput) in integrated networks. However, they often require careful tuning and can face limitations in extremely dynamic or large-scale SAGIN settings. These challenges have prompted increasing interest in intelligent optimization (e.g., reinforcement learning) in recent years, yet the foundational literature on SAGIN still heavily relies on traditional optimization approaches as reviewed above.

2.3.2 Deep Reinforcement Learning based TORA for SAGIN

Deep reinforcement learning (DRL) has emerged as a powerful data-driven alternative to static optimization for SAGIN, enabling agents to learn efficient policies through trial and error exploration in complex, dynamic environments. DRL is particularly well-suited to SAGIN due to its ability to handle high-dimensional state spaces, stochastic environment transitions, and long-term reward dependencies. By continually adapting to changing network conditions, DRL approaches can overcome the limitations of fixed rule-based strategies and trial and error exploration latency, throughput, and energy efficiency.

Zhou et al. [28] introduced a DRL framework for IoT task offloading within SAGIN that explicitly accounts for the energy constraints of UAV nodes. They formulated the scheduling problem as an energy-constrained Markov decision process and developed a deep risk-sensitive reinforcement learning algorithm based on a Deep Q-Network (DQN). The reward function was enhanced with a risk-sensitive term to penalize excessive UAV energy consumption, thereby guiding the agent to respect battery limits. This approach achieved up to 30% average delay reduction in average task processing delay compared to probabilistic (rule-based) baselines, while still satisfying the UAV's energy capacity constraint. The study demonstrated that DRL can dynamically balance energy usage and service latency in resource-constrained aerial nodes, adapting to IoT task arrival dynamics in real time.

Wang et al. [29] extended the DRL paradigm by proposing a distributed multi-agent framework for SAGIN storage resource management. In their design, multiple DRL agents are deployed at edge nodes (e.g., satellite gateways, aerial base stations), each observing local network conditions and training independently in a decentralized manner. To enable cooperative behavior without central coordi-

nation, the authors introduced an inter-agent communication module based on message passing, allowing agents to periodically share summarized state information. Through this mechanism, the distributed agents learned coordinated policies for data storage and offloading across the space-air-ground segments. Simulation results showed an approximately 18.15% improvement in in resource utilization, along with a higher task request acceptance rate, compared to other allocation schemes. Notably, the distributed DRL approach exhibited robust adaptability to changing resource conditions, a design particularly beneficial for large-scale SAGIN deployments where full state observability and centralized control are infeasible.

Subsequent research has further advanced DRL techniques in SAGIN. For example, Huang et al. [30] presented a multi-agent DRL approach for joint task offloading and resource allocation in a hybrid cloud/edge computing SAGIN environment with multiple LEO satellites and UAVs. By employing a novel hybrid action-space algorithm, their framework effectively reduced both energy consumption and end-to-end latency, outperforming conventional rule-based optimization methods in dynamic and complex network scenarios. Similarly, Jia et al. [31] developed a service-oriented SAGIN architecture that integrates edge intelligence (AI at the network edge) to enhance communication, computing, sensing, and storage capabilities. They proposed a DRL-based resource allocation and computation offloading algorithm within this 6G-oriented SAGIN framework. The proposed scheme achieved significant reductions in overall system cost compared to existing heuristics, demonstrating efficient resource management for resource-constrained users under dynamic service demands. These studies highlight the potential of DRL to intelligently orchestrate heterogeneous SAGIN resources, offering adaptive and scalable solutions that improve performance metrics (e.g., delay, energy efficiency, cost) beyond what static or purely centralized approaches can accomplish. Through sophisticated reward design and possibly limited inter-agent communication, DRL-based methods enable SAGIN to autonomously learn

how to balance trade-offs between latency, throughput, and energy in realtime, thereby greatly enhancing network responsiveness and robustness.

2.3.3 GNN-Augmented DRL in TORA of SAGIN

Conventional DRL methods often treat the network environment as a flat input-output mapping, failing to leverage the rich topological structure of SAGIN. To address this limitation, recent studies have integrated Graph Neural Networks (GNNs) with DRL to capture the relational inductive biases and topology of SAGIN environments. In particular, Cai et al. [32] proposed a Graphic DRL (GDRL) framework that represents the network state as a dynamic graph, where nodes correspond to computational entities (satellites, UAVs, ground stations) and edges signify communication or task-offloading links. A GNN module processes this graph-structured state to extract features such as cluster density, network connectivity patterns, and role-based node embeddings (distinguishing, for example, gateway nodes from simple relays). These graph-derived features are then fed into an actor-critic DRL architecture for decision-making. This GDRL approach led to notable performance gains – latency reduced by up to 20% and more consistent reward outcomes under dynamic network conditions – compared to traditional DRL acking graph-awareness mechanisms. The improvement is attributed to the DRL agent's ability to learn the network's structural characteristics, resulting in decisions that better adapt to changes in topology and traffic.

Building on the success of GDRL, other researchers have explored alternative GNN-augmented DRL designs for SAGIN. Wang et al. [33] introduced an attention-based GNN within the DRL agent to more effectively weigh the importance of different node connections, thereby capturing time-varying connectivity patterns in mobile edge computing scenarios. In parallel, Hu et al. [34] developed a spatio-temporal graph convolutional network (ST-GCN) combined with DRL to handle

mobility-induced topology shifts – for instance, the frequent connect/disconnect events in a terahertz-band space network. By incorporating temporal windowing, these models could predict future network states, enabling the agent to anticipate events like congestion build-up or imminent handovers. This predictive capability enabled proactive resource reallocation and task offloading decisions, further improving network performance and resilience under dynamic conditions. Both the attention-based GNN and the ST-GCN approaches demonstrated improved efficiency in simulations, as the DRL agents could foresee and mitigate performance drops before they occur (e.g., rerouting or load balancing ahead of a link becoming overloaded).

Further extending the GNN-DRL paradigm, Song et al. [35] proposed an intent-driven network orchestration framework for SAGIN. In their approach, a DRL agent augmented by GNN state representation is used to map high-level service requirements or intents (such as guaranteed bandwidth, low latency, or priority for certain applications) onto low-level configurations spanning heterogeneous network domains. The GNN captures the cross-domain dependencies in SAGIN (space, air, ground segments), while the DRL component learns how to satisfy the given intent by appropriate resource allocation and routing. This intent-based DRL+GNN agent can dynamically adjust resources – for example, allocating more satellite link capacity or edge computing power – to meet the specified service goals across the integrated network. By bridging high-level intents with on-the-fly control decisions, Song et al.'s work highlights a new level of autonomy in SAGIN management. It demonstrates that incorporating GNN-derived topological insights in DRL not only improves raw performance metrics such as latency or throughput, but also facilitates policy-driven orchestration that aligns network behavior with user-defined objectives across the space–air–ground continuum.

2.3.4 Limitations of Static and Homogeneous Graph Representations

Despite these advances, most existing GNN-enhanced DRL models rely on static or homogeneous graph assumptions. In practice, SAGIN exhibits dynamic and heterogeneous characteristics: LEO satellites move rapidly in orbit, UAVs follow task-specific trajectories, and ground stations are unevenly distributed. Additionally, link properties (e.g., delay, jitter, bandwidth) vary significantly across satellite-to-ground, air-to-ground, and inter-satellite channels.

Static graphs fail to capture these temporal dynamics, leading to suboptimal decisions when real-time adaptation is required. Homogeneous graphs oversimplify the network by treating all nodes and links as equal, ignoring the functional heterogeneity of nodes (e.g., sensing-only IoT devices vs. compute-enabled UAVs). These limitations reduce the generalizability and scalability of existing methods in realistic SAGIN environments.

2.3.5 Summary

In this chapter, we reviewed the recent developments in resource allocation and task offloading schemes in Space–Air–Ground Integrated Networks (SAGIN). The reviewed methods span three categories: traditional optimization approaches, deep reinforcement learning (DRL) methods, and GNN-enhanced DRL methods.

Traditional optimization methods model the resource allocation problem as a one-shot, often non-convex, optimization problem. Techniques such as convex relaxation, dual decomposition, and evolutionary algorithms have been applied to these problems. These approaches provide strong theoretical guarantees and are often suitable for small-scale or static networks. However, their applicability in dy-

namic and large-scale SAGIN is limited by computational complexity and reliance on complete, static network state information. Most notably, they cannot handle frequent topological changes or adapt rapidly to mobility and link fluctuations.

Deep reinforcement learning (DRL) approaches overcome these limitations by learning policies that map environment states to actions through trial and error. They are well-suited for high-dimensional, stochastic environments such as SAGIN. DRL-based models support online decision-making and achieve better generalization under changing network states. However, they often fail to incorporate the underlying topological structure of the network, instead treating the environment as a flat observation space, which hinders scalability and interpretability.

GNN-augmented DRL methods address this shortcoming by modeling SAGIN as a graph-structured environment. These models utilize graph neural networks (GNNs) to extract structural features such as node connectivity, traffic density, and mobility patterns, which in turn make the DRL agent's decisions more robust against topological variations. Techniques such as attention mechanisms, spatio-temporal modeling, and intent-driven orchestration have further improved performance. Yet, these methods often rely on static or homogeneous graphs, neglecting the heterogeneity and temporal evolution inherent in real-world SAGIN systems.

Table 2.1 summarizes the representative works from each category, comparing their design goals, methods, strengths, and key limitations.

Table 2.1: Summary of Key Literature in SAGIN Resource Allocation and Offloading

Work Category	**Design Goal**	**Methodology**	**Limitations**
Traditional Optimization	Minimize latency or maximize throughput under resource constraints	Convex optimization, dual decomposition, genetic algorithms, etc.	Static assumptions, centralized control, high complexity, poor adaptability
DRL-Based Methods	Online adaptive decision-making in dynamic environments	DQN, actor-critic methods, multi-agent DRL, reward shaping	Ignores topology, difficult convergence in large state spaces
GNN-Enhanced DRL	Structure-aware and scalable decision-making	GDRL, ST-GCN, intent-based DRL, attention-based GNNs	Static homogeneous graphs, limited modeling of mobility and heterogeneity

In summary, while significant progress has been made in designing intelligent offloading and resource allocation policies for SAGIN, most existing works still suffer from crucial limitations. Particularly, the assumption of static or homogeneous network graphs constrains the adaptability and realism of current models. Real-world SAGINs exhibit highly dynamic and heterogeneous properties—LEO satellites move rapidly, UAVs follow mission-specific flight paths, and different network segments (space, air, ground) feature vastly distinct link characteristics.

To address these challenges, our work proposes a novel resource allocation framework based on a **heterogeneous dynamic graph neural network**. This approach explicitly models node and link heterogeneity, captures temporal dependencies, and integrates these features into a DRL architecture for high-quality decision-making in real-time. Our proposed method builds on the GDRL foundation introduced by Cai et al. [32], and extends it to a solution with stronger **generalization** and **robustness** for real-world SAGIN deployments.

In the context of this thesis, *generalization* refers to the ability of the learned policy to remain effective across a wide spectrum of unseen SAGIN configurations, including different satellite trajectories, user distributions, traffic patterns, and environmental dynamics. This capability arises from the dynamic heterogeneous graph representation, which removes the fixed-topology assumption in [32] and allows the policy network to process arbitrary and time-varying connectivity patterns. As a result, the trained model does not overfit to a specific topology or static role assignment, and maintains performance under topology shifts encountered during deployment.

Robustness in our framework refers to the stability of decision-making under rapid link fluctuations, intermittent connectivity, and role changes of non-terrestrial nodes. The proposed graph-update mechanism and the Dynamic Node Functional Switching Mechanism (DFSM) jointly contribute to this robustness: the graph updates capture fast-changing link states without requiring model retraining, while DFSM prevents suboptimal actions by constraining the feasible action set in accordance with each node's instantaneous functional state. Together, these components allow the system to produce reliable offloading and allocation decisions even when the network undergoes abrupt or unpredictable changes—conditions where the baseline method in [32] degrades significantly due to its assumption of relatively stable graph structure.

By explicitly clarifying these notions of generalization and robustness, we highlight how the proposed DHGNN+DFSM framework offers improved adaptability and stability in realistic SAGIN operational environments.

Chapter 3

Dynamic Graph–Aware TORA

This chapter develops a *Dynamic Heterogeneous Graph Neural Network (DHGNN)* for joint task offloading and resource allocation in time-varying Space–Air–Ground Integrated Networks (SAGIN). We begin by formalizing the SAGIN system model and notation in a manner consistent with the rest of the thesis—defining the node sets $\mathcal{U}$ (UEs), $\mathcal{H}$ (HAPS), and $\mathcal{L}$ (LEOs), the slotted time index t, the wireless channel $h_{u,k}^{X}$ and rate models for delay-tolerant and delay-sensitive traffic, as well as the energy accounting. On top of this model, we construct a *dynamic heterogeneous graph* whose adjacency $\mathbf{A}^t$ and relation-specific edge attributes encode both link quality and resource availability, thereby capturing structural heterogeneity (UE/HAPS/LEO) and temporal evolution due to mobility. A heterogeneity-aware GNN backbone performs message passing over the typed subgraphs and produces state embeddings that are fused with non-graph variables to parameterize a policy for multi-objective optimization (latency and energy). A periodic topology-update mechanism maintains $\mathbf{A}^t$ as nodes move, balancing accuracy and computational cost. The remainder of the chapter details the system model, the graph construction and edge-weighting strategy, the DHGNN architecture and training pipeline, and a comprehensive evaluation against static and homogeneous graph baselines

demonstrating gains in cumulative return, inference latency, and power consumption.

3.1 SAGIN system model

The *SAGIN* refers to a networking architecture that integrates satellite communication systems (space segment), airborne platforms (air segment, e.g. high-altitude platforms, drones/UAVs), and terrestrial networks (ground segment) into a unified framework. This three-layer integration is designed to provide seamless global coverage and ubiquitous connectivity by leveraging the complementary advantages of each segment. For instance, satellites (particularly low Earth orbit (LEO) constellations) offer ultra-wide coverage, aerial platforms provide flexible deployment and line-of-sight links, and ground networks deliver high-capacity local access [6], [7]. By jointly utilizing space, air, and ground nodes, SAGIN can serve a broad range of scenarios, including remote or hard-to-reach areas where terrestrial infrastructure is lacking, thus narrowing the digital divide. In this thesis, we consider a multi-layer SAGIN consisting of terrestrial user equipments (**UEs**) , high-altitude platform stations (**HAPSs**), and low Earth orbit LEO satellites. The network topology evolves dynamically due to node mobility and varying channel conditions. Fig 2.1 shows the overview of the SAGIN system.

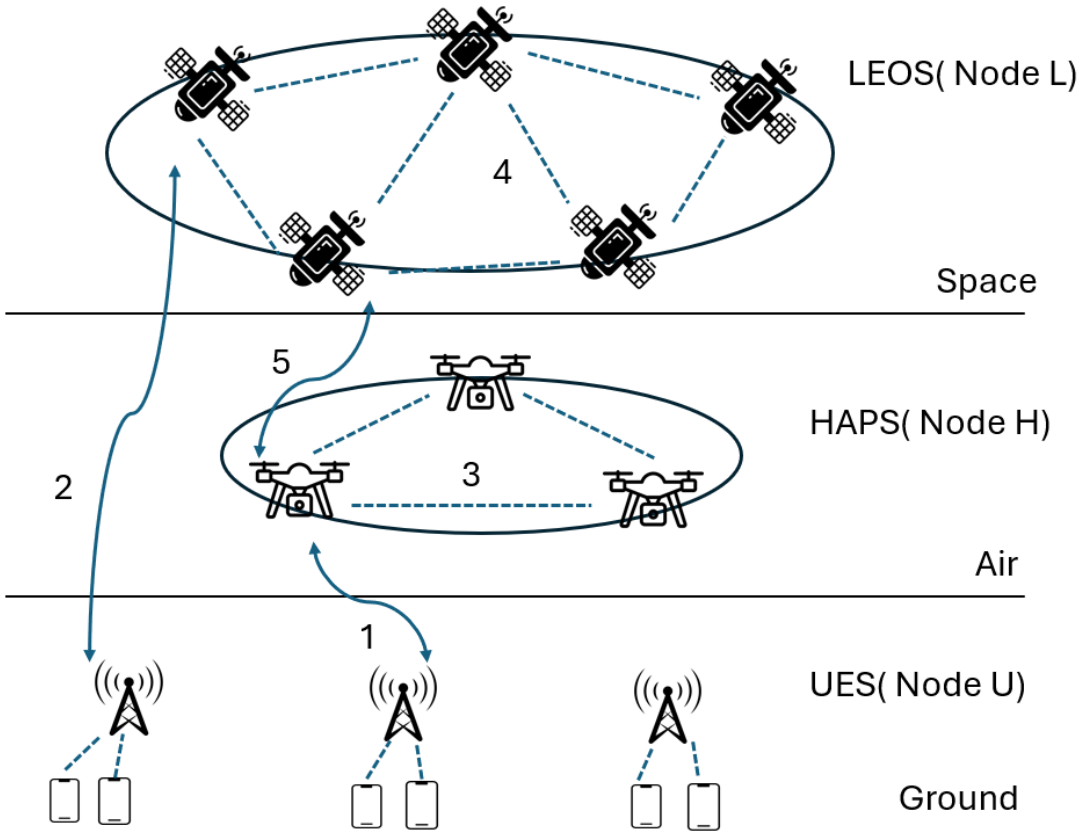

Figure 3.1: Overview of the SAGIN System

3.1.1 Network Topology and Task Generation

The sets of nodes are defined as follows:

$$\begin{aligned}\mathcal{L} &= \{1, 2, \ldots, S\} \quad \text{for LEO satellites,} \\ \mathcal{H} &= \{1, 2, \ldots, A\} \quad \text{for HAPS nodes,} \\ \mathcal{U} &= \{1, 2, \ldots, U\} \quad \text{for UEs.}\end{aligned}$$

Each UE $u \in \mathcal{U}$ periodically generates two types of computation tasks: delay-sensitive tasks δ_u, which have strict latency requirements, and delay-tolerant tasks τ_u, which allow more flexible delays. Time is divided into a short-term period $\mathcal{T}_Short = \{1, 2, \ldots, S\}$ for handling delay-sensitive tasks and a long-term period $\mathcal{T}_Long = \{1, 2, \ldots, L\}$ for delay-tolerant tasks.

3.1.2 Wireless Channel Model

We consider five primary wireless link types: ground-to-air, ground-to-space, inter-HAPS, inter-satellite, and air-to-space. Each HAPS is equipped with a downward-facing Uniform Planar Array (UPA) consisting of $N_H = N_H^x \times N_H^y$ antenna elements, and similarly, each LEO satellite features a UPA with $N_L = N_L^x \times N_L^y$ elements, where $N_H^x, N_H^y, N_L^x, N_L^y$ denote the number of antenna elements in the horizontal and vertical directions respectively.

The channel gain for a wireless link from UE u to a node of type $X \in \{\mathrm{L}, \mathrm{H}\}$ (LEO or HAPS) over subchannel k is modeled as:

$$h_{u,k}^X = \sqrt{\phi_u^X} \left(\frac{c}{4\pi D_u^X f_k^X} \right)^{\theta_{Xu}} \zeta_{Xu,k} \mathbf{v}_u^X, \tag{3.1}$$

where c is the speed of light, ϕ_u^X represents the effective antenna gain combining UE

and node gains, D_u^X is the distance between UE u and node X, f_k^X is the carrier frequency for subchannel k, θ_{Xu} captures attenuation effects, $\zeta_{Xu,k}$ is a Rician fading coefficient, and $\mathbf{v}_u^X \in \mathbb{C}^{N_X \times 1}$ is the UPA response vector with $N_X = N_H$ if $X = H$ and N_L if $X = L$.

Links such as inter-satellite and inter-HAPS operate under quasi-vacuum conditions and are modeled as additive white Gaussian noise (AWGN) channels.

In the wireless channel model of the system, Equation (2.1) characterizes signal propagation including path loss and fading. This formula originates from classical wireless communication theory and experimental channel modeling. Large-scale propagation loss typically follows a distance-based attenuation law (e.g., a power-law path loss model), which is then combined with a random small-scale fading factor to capture multipath propagation effects [36].

For the small-scale fading term, two well-known models are usually considered: *Rayleigh fading* and *Rician fading*. Rayleigh fading applies when no dominant line-of-sight (LoS) path exists, suitable for urban terrestrial links [36]. Rician fading models channels where a strong LoS component coexists with scattered multipath components and is common in satellite and aerial links [36]. The fading component in Equation (1) accounts for these propagation conditions.

Equation (2.1) thus combines large-scale path loss and small-scale multipath fading, foundational in wireless channel modeling [36]. Key parameters—including the path loss exponent, the Rician Kfactor, and antenna gains—are typically obtained from experimental measurements or standardized channel models, and they underpin the calculation of instantaneous channel gains used later in rate and energy computations.

3.1.3 Data Transmission Rates

For delay-tolerant tasks τ_u, the achievable transmission rate via node X on sub-channel k is:

$$R^X_{\tau,u,k} = \log_2 \left(1 + \frac{P_{\tau,u}|h^X_{u,k}|^2}{\sigma^2}\right), \tag{3.2}$$

where $P_{\tau,u}$ is the allocated transmit power, and σ^2 is noise power.

For delay-sensitive tasks δ_u, the finite blocklength effect is considered, and the transmission rate is modeled as:

$$R^X_{\delta,u,k} = \frac{B}{\ln 2}\left[\ln\left(1 + \frac{P_{\delta,u}|h^X_{u,k}|^2}{\sigma^2 B}\right) - \sqrt{\frac{1}{T_b B}} f_Q^{-1}(\epsilon_{\delta,u})\right], \tag{3.3}$$

where B is the subcarrier bandwidth, T_b is block duration, $f_Q^{-1}(\cdot)$ is the inverse Q-function, and $\epsilon_{\delta,u}$ is the target decoding error probability.

Equation (3.3) models the achievable transmission rate under finite blocklength constraints, reflecting recent advances in information theory [37]. In contrast to the classical Shannon capacity formula, which assumes infinitely long codewords, finite blocklength theory establishes bounds on coding rates for short packets under a given nonzero error probability . This is crucial for URLLC (ultra-reliable low-latency communication) scenarios common in 5G/6G [38].

The formula in Equation (3.3) derives from the normal approximation in channel coding theory, incorporating the channel dispersion term and the inverse Gaussian Q-function [37]. This model accurately quantifies the rate penalty due to short codes and desired reliability, making it suitable for delay-sensitive tasks in SAGIN.

3.1.4 Computation Task Model

Each computation task generated by UE u at time t is described by the tuple:

$$\Psi_u(t) = \{\chi_u(t), \sigma_u(t), \rho_u(t), \delta_u(t)\},$$

where $\chi_u(t)$ is the required CPU cycles per bit, $\sigma_u(t)$ is the input data size (bits), $\rho_u(t)$ is the output data size (bits), and $\delta_u(t)$ is the maximum tolerable delay.

Task generation follows a probability π_u per time slot. Execution decisions are denoted by binary variables:

$$\eta_u(t) = \begin{cases} 1, & \text{local execution} \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

$$\beta_{u,j}(t) = \begin{cases} 1, & \text{offloaded to node } j \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

$$\gamma_{u,j,k}(t) = \begin{cases} 1, & \text{task relayed via node } j \text{ to } k \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

In other word,

$$\alpha_u(t) + \sum_{j=1}^{L+H} \beta_{u,j}(t) + \sum_{j=1}^{L+H} \sum_{k=1}^{L+H+1} \xi_{u,j}\, \xi_{j,k}\, \gamma_{u,j,k}(t) \leq 1. \tag{3.7}$$

The network connectivity is represented by the adjacency matrix $\mathbf{A} \in \{0,1\}^{(U+L+H)\times(U+L+H)}$, partitioned as follows: rows 1 to U correspond to UEs, $U+1$ to $U+L$ for LEOs, and $U+L+1$ to $U+L+H$ for HAPS. The execution decision is determined using binary variables: $\alpha_u(t) = 1$ for local execution; $\beta_{u,j}(t) = 1$ if offloaded to node j (where $j \in \mathcal{L} \cup \mathcal{H}$); and $\gamma_{u,j,k}(t) = 1$ if the task is relayed via node j to

node k. We implement a multi-objective optimization approach that balances task completion latency with energy efficiency, addressing the practical constraints of SAGIN deployments.

3.1.5 Delay Model

In this subsection we make explicit how the completion latency of each task is computed under different execution options (local, one–hop offloading, and two–hop relaying). This clarifies the multi–objective formulation in Section **??** and the "actual processing latency" term used in the reward definition.

Recall that UE u generates at time slot t a task $\Psi_u(t) = \{\chi_u(t), \sigma_u(t), \rho_u(t), \delta_u(t)\}$, where $\chi_u(t)$ is the required CPU cycles per bit, $\sigma_u(t)$ and $\rho_u(t)$ are the input and output sizes (bits), and $\delta_u(t)$ is the end–to–end latency requirement. The execution decision is encoded by $\eta_u(t)$ (local execution), $\beta_{u,j}(t)$ (one–hop offloading to node $j \in \mathcal{L} \cup \mathcal{H}$), and $\gamma_{u,j,k}(t)$ (two–hop relaying via j to k), with the single–choice constraint (4.4).

Processing delay. Let F_u denote the local CPU frequency of UE u (cycles/s), and F_j^{cpu} the aggregate CPU capacity of node j. When a task from u is processed locally, its processing delay is $\chi_u(t)\sigma_u(t)/F_u$. When it is offloaded to a remote node, a fraction $\theta_{j,u}(t) \in (0, 1]$ of F_j^{cpu} is allocated to this task. The processing delay for UE u at slot t is therefore

$$L_{\text{proc},u}(t) = \eta_u(t)\,\frac{\chi_u(t)\,\sigma_u(t)}{F_u} + \sum_{j\in\mathcal{L}\cup\mathcal{H}} \beta_{u,j}(t)\,\frac{\chi_u(t)\,\sigma_u(t)}{\theta_{j,u}(t)\,F_j^{\text{cpu}}} + \sum_{j\in\mathcal{L}\cup\mathcal{H}}\sum_{k\in\mathcal{L}\cup\mathcal{H}} \gamma_{u,j,k}(t)\,\frac{\chi_u(t)\,\sigma_u(t)}{\theta_{k,u}(t)\,F_k^{\text{cpu}}}. \tag{3.8}$$

The three terms correspond to local execution, one–hop offloading, and two–hop relaying, respectively.

Communication delay. We separate uplink transmission of the input $\sigma_u(t)$ and downlink transmission of the result $\rho_u(t)$. Let $R^X_{u\to n}(t)$ denote the achievable rate (either $R^X_{\tau,u,k}$ or $R^X_{\delta,u,k}$ depending on the task type) from UE u to node n in layer $X \in \{\mathrm{L}, \mathrm{H}\}$ at slot t, and $R^Y_{n\to m}(t)$ the rate from node n to node m in layer Y. Then the communication delay for UE u is

$$L_{\mathrm{comm},u}(t) = \eta_u(t) \underbrace{0}_{\text{local execution}} + \sum_{j\in\mathcal{L}\cup\mathcal{H}} \beta_{u,j}(t)\left(\frac{\sigma_u(t)}{R^{\mathrm{type}(j)}_{u\to j}(t)} + \frac{\rho_u(t)}{R^{\mathrm{type}(j)}_{j\to u}(t)}\right) + \sum_{j\in\mathcal{L}\cup\mathcal{H}}\sum_{k\in\mathcal{L}\cup\mathcal{H}} \gamma_{u,j,k}(t)\left(\frac{\sigma_u(t)}{R^{\mathrm{type}(j)}_{u\to j}(t)} + \frac{\sigma_u(t)}{R^{\mathrm{type}(k)}_{j\to k}(t)} + \frac{\rho_u(t)}{R^{\mathrm{type}(j)}_{k\to j}(t)} + \frac{\rho_u(t)}{R^{\mathrm{type}(u)}_{j\to u}(t)}\right), \tag{3.9}$$

where type$(\cdot)$ returns the layer (LEO/HAPS/UE) of the node so that the appropriate rate expression from Section 4.1 is used. For local execution there is no communication delay.

Handover and propagation delay. When the serving LEO/HAPS for a UE changes due to mobility, the task experiences an additional handover delay. Let $N^{\mathrm{ho}}_u(t)$ be the number of handovers involving UE u at slot t (typically 0 or 1) and T_{ho} the per–handover delay budget. We model the handover delay as

$$L_{\mathrm{ha},u}(t) = N^{\mathrm{ho}}_u(t)\, T_{\mathrm{ho}}. \tag{3.10}$$

Large–scale propagation delay on long space links is already embedded into the effective rate terms $R^X_{\cdot}(t)$; (3.10) thus captures only additional signaling and re–association overhead.

Total completion delay and task types. Combining the above components, the end–to–end completion delay of a task from UE u at slot t is

$$L_{\mathrm{tot},u}(t) = L_{\mathrm{proc},u}(t) + L_{\mathrm{comm},u}(t) + L_{\mathrm{ha},u}(t). \tag{3.11}$$

For delay–sensitive tasks δ_u we define

$$L_{\delta,u}(t) = \omega_\delta \, L_{\mathrm{tot},u}(t), \quad \text{while for delay–tolerant tasks } \tau_u \text{ we set} \quad L_{\tau,u}(t) = \omega_\tau \, L_{\mathrm{tot},u}(t), \tag{3.12}$$

with weighting coefficients $\omega_\delta > \omega_\tau > 0$ reflecting their relative delay criticality. The aggregated system delay in slot t then follows (3.11), where $L_{\delta,u}(t)$ and $L_{\tau,u}(t)$ are chosen according to the realized task type.

In the sequel, the "actual completion latency" (or "actual processing latency") used in the reward function is precisely $L_{\mathrm{tot},u}(t)$ in (3.11), i.e., the realized end–to–end delay including processing, communication, and handover components.

3.1.6 Energy Consumption Model

The communication energy model assumes transmit energy proportional to power and transmission time, linked to achievable rate and channel conditions via Shannon's law [36]. The computation energy model is based on processor power consumption scaling with CPU frequency and voltage, following CMOS circuit principles and dynamic voltage-frequency scaling [39]. Experimental studies validate these models for mobile and edge computing platforms [40], [41].

The combined energy models underpin the trade-offs in task offloading strategies within SAGIN, guiding resource allocation to balance latency and energy efficiency. Energy consumption consists of transmission and computation energy components.

The transmission energy for UE u is modeled as:

$$E_{\text{user}}^{\text{tx}} = \frac{P_u \cdot S_u}{R_{u,k}} + \varepsilon, \tag{3.13}$$

where P_u is UE transmission power, S_u is data size, $R_{u,k}$ is transmission rate on subchannel k, and ε avoids division by zero.

The node transmission and computation energy are given by:

$$E_{\text{node}}^{\text{tx}} = \frac{P_{\text{node}} \cdot S_u}{R_{u,k}} + \varepsilon, \tag{3.14}$$

$$E_{\text{node}}^{\text{cpu}} = \kappa_{\text{node}} \cdot (S_u \cdot O_u), \tag{3.15}$$

where P_{node} is node transmission power, κ_{node} is computation energy efficiency, and O_u denotes computational complexity per bit.

The total energy consumption by UE u is the sum of the above components:

$$E_u^{\text{total}} = E_{\text{user}}^{\text{tx}} + E_{\text{node}}^{\text{tx}} + E_{\text{node}}^{\text{cpu}}. \tag{3.16}$$

3.2 The Proposed Dynamic Heterogeneous Graph Learning Framework for SAGIN Optimization

This section presents our DHGNN framework for multi-objective resource optimization in SAGIN. We introduce novel techniques to address the challenges of node mobility, heterogeneous network characteristics, and competing optimization objectives , as illustrated in Figure 3.2. Algorithm 1 details our approach, which integrates periodic topology updates with heterogeneous GNN-based learning for

efficient resource allocation . The framework operates in a closed-loop manner, continuously adapting to evolving network conditions through periodic topology updates and learning-based decision making. The algorithm operates in two distinct phases: First , construct an initial heterogeneous graph by generating an adjacency matrix that encodes node positions, resource levels, and inter-node relationships, and then derive the corresponding heterogeneous edge index. Next , at each time step, extract state embeddings via a heterogeneous GNN to guide the policy network's resource allocation decisions. Every K iterations (lines 10–14), update the graph topology to capture dynamic changes in node mobility, channel conditions, and resource availability.

Algorithm 1 Dynamic Heterogeneous GNN for SAGIN Resource Allocation

1: **Initialize:** Node positions $\mathbf{P}_U, \mathbf{P}_L, \mathbf{P}_H$; Resources $\mathbf{R}$; Requests $\mathbf{Q}$; $t \leftarrow 0$
2: $\mathbf{A}^0 \leftarrow \text{GenerateHeteroAdjacency}(U, L, H)$
3: $\mathbf{A}^t \leftarrow \text{update\_dynamic\_adj\_matrix}(\mathbf{A}^0, \mathbf{Q}, \mathbf{P}, \mathbf{R})$
4: $\mathcal{E} \leftarrow \text{generate\_hetero\_edge\_index}(\mathbf{A}^t, U, L, H)$
5: **while** not terminated **do**
6: $\quad f_t \leftarrow \text{HeteroGNNExtractor}(s_t, \mathcal{E})$
7: $\quad a_t \leftarrow \text{PolicyNetwork}(f_t)$
8: $\quad$ Execute a_t, receive r_t and s_{t+1}
9: $\quad t \leftarrow t + 1$
10: $\quad$ **if** $t \bmod K = 0$ **then**
11: $\quad\quad$ Update positions: $\mathbf{P}_U, \mathbf{P}_L, \mathbf{P}_H$
12: $\quad\quad$ Update channel conditions and resources $\mathbf{R}$
13: $\quad\quad \mathbf{A}^t \leftarrow \text{update\_dynamic\_adj\_matrix}(\mathbf{A}^0, \mathbf{Q}, \mathbf{P}, \mathbf{R})$
14: $\quad\quad \mathcal{E} \leftarrow \text{generate\_hetero\_edge\_index}(\mathbf{A}^t, U, L, H)$
15: $\quad$ **end if**
16: **end while**
17: **Update** model parameters

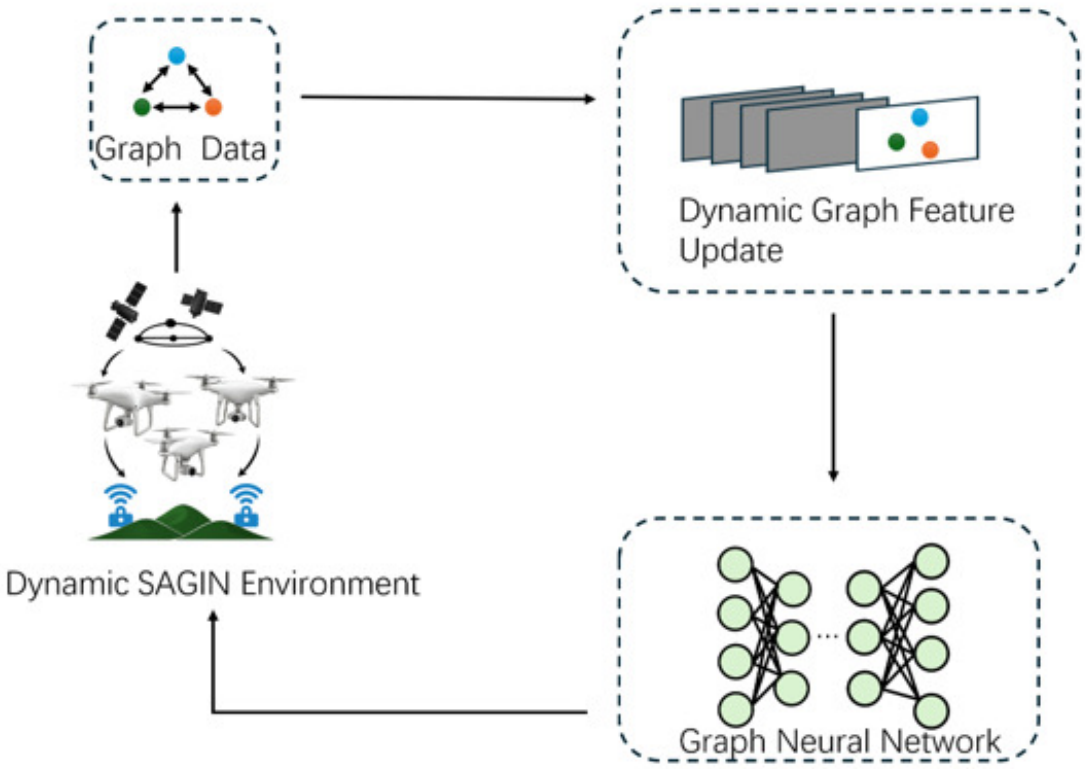

Figure 3.2: DHGNN interacted with the environment

3.2.1 Dynamic Heterogeneous Graph Implementation

We propose a dynamic heterogeneous graph implementation that explicitly captures the unique node types and evolving relationships within SAGIN.

We construct a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ where each node is assigned a type $\tau(v) \in \{user, HAPS, LEO\}$. Each node type maintains its own distinct feature representation:

$$\mathbf{h}_v^{(0)} = \begin{cases} \mathbf{f}_{user}(v) & \text{if } \tau(v) = user, \\ \mathbf{f}_{HAPS}(v) & \text{if } \tau(v) = HAPS, \\ \mathbf{f}_{LEO}(v) & \text{if } \tau(v) = LEO. \end{cases} \tag{3.17}$$

To enable efficient GNN processing, we transform the weighted adjacency matrix into relation-specific sparse representations.

$$\mathcal{E}_{(src,rel,dst)} = \{\mathbf{edge\_index}, \mathbf{edge\_attr}\}. \tag{3.18}$$

This supports five primary relation types: $(user, connects, LEO)$, $(user, connects, HAPS)$, $(LEO, connects, LEO)$, $(HAPS, connects, HAPS)$, and $(node, self\_loop, node)$. In our approach, connections are divided by heterogeneous node types to create subgraphs with their own adjacency matrices instead of constructing a single large adjacency matrix.

3.2.2 DHGNN Edge Weight Calculation

We calculate DHGNN edge weights by combining channel quality and resource availability:

$$w_t^{i,j} = \alpha \cdot |h_t^{i,j}| \cdot \frac{r_t^i + r_t^j}{2}, \tag{3.19}$$

where $|h_t^{i,j}|$ is the channel gain at time t, as defined in (**??**), and r_t^i and r_t^i represent the available resources of node i and j at time t. The weights are normalized for numerical stability, and we have

$$\mathbf{A}_{i,j}^t = \frac{w_t^{i,j}}{\max_{i,j}(w_t^{i,j})} \cdot \mathbf{A}_{i,j}^0. \tag{3.20}$$

The Figure 3.3 shows the overview of the mechanism with time step variance.

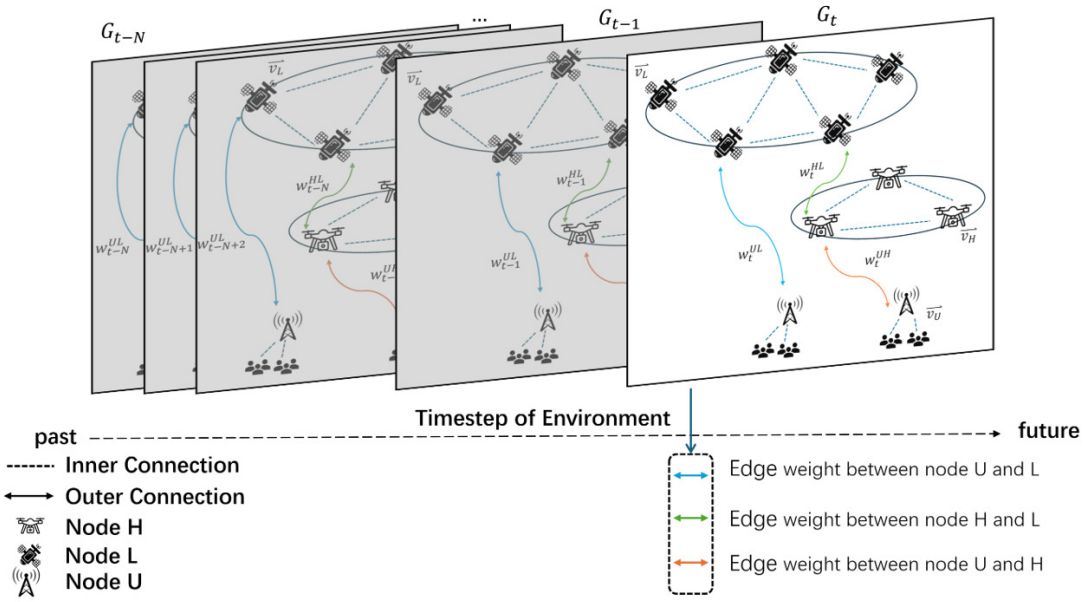

Figure 3.3: The dynamic heterogeneous graph update mechanism

3.2.3 Heterogeneous Graph Neural Network

As shown in Figure 3.4, our heterogeneous GNN architecture processes the dynamic SAGIN topology through specialized pathways for each node type. The network separates input observations into non-graph state variables and node-specific attributes for users, LEO satellites, and HAPS nodes. Each node type is initially projected into a shared hidden space via linear layers. Rather than building a single global graph, we construct type-specific subgraphs with dynamic edge indices and attributes that capture the current channel conditions and resource availability.

Message passing is performed by GATv2Conv layers on these heterogeneous subgraphs, with layer normalization and residual connections to refine node features. Global average pooling aggregates these node outputs, which are then concatenated with the non-graph state and fed into a final feedforward network to produce the overall feature vector. This design efficiently adapts to DHGNN topologies and heterogeneous relationships.

3.2.4 Position Update Process

A key innovation in our approach is the periodic topology update that adapts to node mobility. This mechanism executes every K time slots (when $t \mod K = 0$), efficiently balancing adaptation with computational overhead as shown in Figure 3.3. When the update condition is met, all node positions are updated according to their respective mobility models by (3.19):

$$\text{if } t \bmod K = 0 : \begin{cases} \mathbf{W}_t^{UL} = \text{UpdateUsers-LEOS}, \\ \mathbf{W}_t^{HL} = \text{UpdateHAPS-LEOS}, \\ \mathbf{W}_t^{UH} = \text{UpdateUser-HAPS}. \end{cases} \tag{3.21}$$

Based on [42], the movements of HAPS and users are modeled as drifts in a specific region, while LEO satellites follow a simplified kinematic model. The position updates are given by

$$x_i^{(t+1)} = (x_i^{(t)} + v_{\text{orbit}} \cdot \Delta t) \bmod R_{\text{region}}, \tag{3.22}$$

$$y_i^{(t+1)} = h_{\text{base}} + V \cdot \sin\left(\frac{x_i^{(t+1)}}{R_{\text{region}}} \cdot 2\pi\right), \tag{3.23}$$

where $x_i^{(t)}$ and $y_i^{(t)}$ represent the horizontal position and altitude of the i-th LEO satellite at time step t, respectively. v_{orbit} is the orbital velocity, Δt is the time interval between consecutive updates, and R_{region} specifies the horizontal extent of the simulation area.h_{base} is the base altitude with an amplitude variation V. The modulo operation ensures cyclical movement within the defined region.

3.2.5 Reinforcement Learning

In our proposed framework, the reinforcement learning agent interacts with the dynamic SAGIN environment in an iterative on-policy manner. At each time step, the agent observes the current state $s \in \mathbb{R}^n$ and uses the dynamic heterogeneous graph feature extractor to generate a latent action vector a. This latent vector encodes high-level decision information extracted from the complex, time-varying network topology.

The latent action vector is divided into two parts: a discrete component a_d and a continuous component a_c. The discrete segment is processed through an encoder $f_d(\cdot)$ to produce binary scheduling decisions, while the continuous segment passes through another encoder $f_c(\cdot)$ to yield modulation parameters for resource allocation. Formally, the mapping performed by the action mapping network (AMN) is

given by

$$\mathrm{AMN}(a) = \Big(f_d(a),\ f_c(a)\Big). \tag{3.24}$$

The resulting actions (a_d, a_c) are applied to the environment, which is modeled as a Markov Decision Process (MDP) with state transition probability $p(s' \mid s, a)$ and reward function $r(s, a)$. To effectively capture the evolving network conditions, the rollout length for our on-policy reinforcement learning strategy is set to 100. The environment saves the status, receives input actions, and returns feedback to the agent; Finally, our joint optimization uses a weighted reward function

$$R(u, k) = \gamma_1(L_u - L_{\mathrm{tot},u}(t) \cdot C_1) - \gamma_2(E^u_{\mathrm{total}} \cdot C_2), \tag{3.25}$$

where L_u is the latency requirement, $L_{\mathrm{tot},u}(t)$ is the total system delay per time slot, E^u_{total} is the total energy consumption and γ_1, γ_2, C_1, and C_2 are weighting factors appropriately scaled to balance latency and energy objectives.

3.3 Simulation Result and Comparison

3.3.1 Experimental Settings

To validate the effectiveness of the proposed **Dynamic Heterogeneous Graph Neural Network (DHGNN)**, we simulate a time-varying SAGIN environment. The scenario comprises $U = 3$ terrestrial users, $H = 16$ high-altitude platform stations HAPS, and $L = 16$ low Earth orbit LEO satellites. Users continuously generate task requests characterized by stochastic computational and transmission requirements.

The simulated space region spans $R_{\mathrm{region}} = 2000\,\mathrm{km}$ with orbital velocity $v_{\mathrm{orbit}} = 7.5\,\mathrm{km/s}$, and LEO base altitude $h_{\mathrm{base}} = 750\,\mathrm{km}$ with sinusoidal variation ampli-

tude $A = 10\,\text{km}$ [42]. The base time granularity is set to $T_b = 10^{-3}\,\text{s}$, and the precision threshold is $\epsilon_{\delta\mu} = 10^{-3}$. The normalized bandwidth is $K = 100\,\text{dBm/Hz}$, and the user-specific penalty weight is $\phi_u^X = 10$.

The agent employs an on-policy reinforcement learning strategy with rollout horizon 100. The GNN model uses hidden dimension $d_{\text{hid}} = 32$, output size $d_{\text{out}} = 32$, multi-head attention with $H = 4$ heads, dropout rate $p = 0.3$, and two-layer MLP with sizes 128 and 64, respectively.

We compare our method against three baseline models:

1.SHOM: Static Homogeneous Graph Model [32], representing conventional fixed-topology RL without node differentiation.

2.SHEM: Static Heterogeneous Graph Model with fixed node types but no temporal dynamics.

3.DHOM: Dynamic Homogeneous Graph Model capturing temporal evolution but lacking node-type diversity.

3.3.2 Policy Learning Performance

Figure 3.5 illustrates the training reward evolution across the four models. DHGNN demonstrates the fastest convergence and achieves the highest average return, outperforming SHOM by **282.08%**, SHEM by **40.32%**, and DHOM by **4.66%**. This confirms the effectiveness of simultaneously incorporating temporal and structural heterogeneity in the graph formulation.

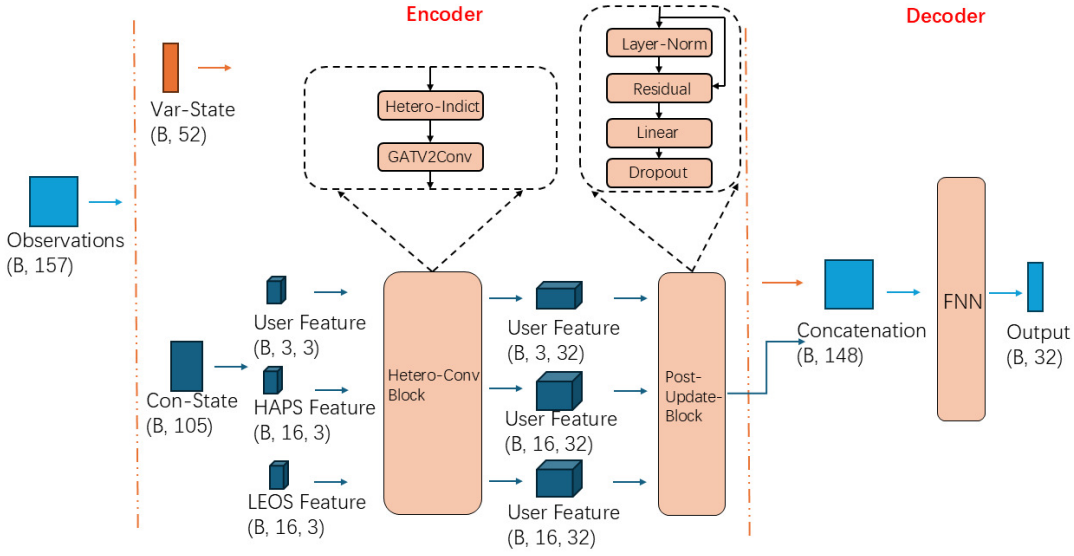

Figure 3.4: The heterogeneous graph neural network

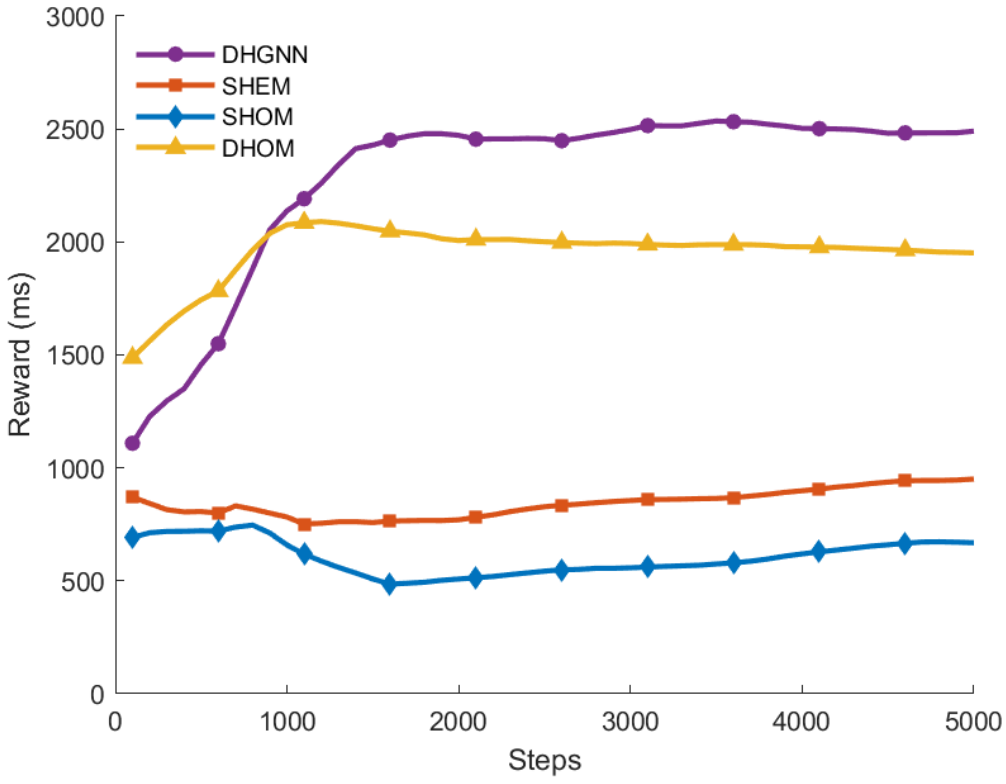

Figure 3.5: Training reward curves of DHGNN versus three baselines.

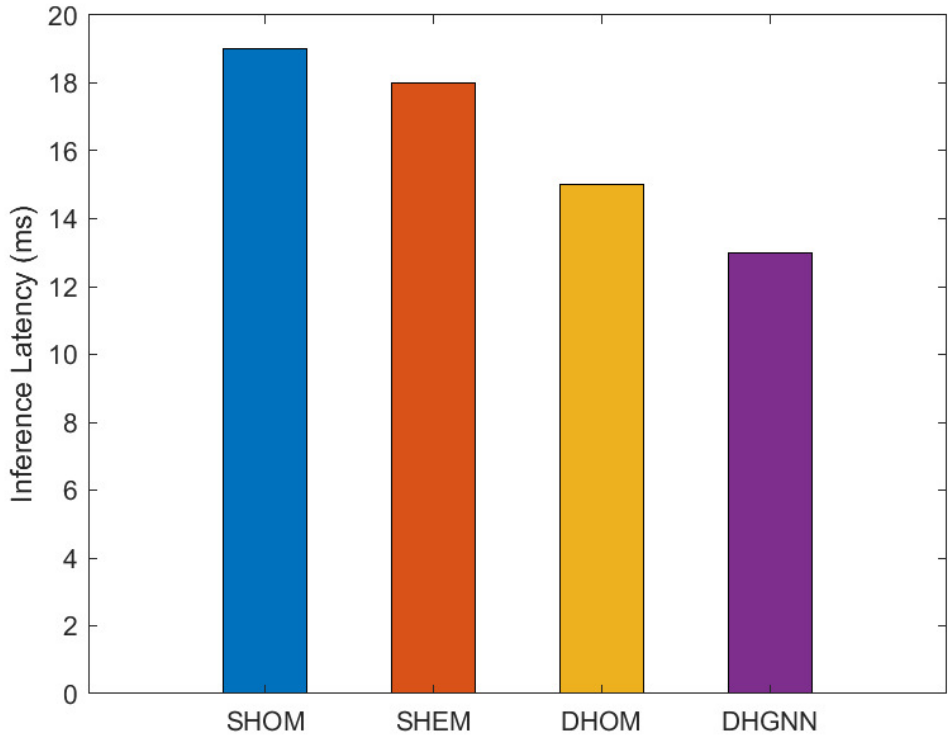

Figure 3.6: Per-step inference latency (ms) comparison.

3.3.3 Inference Latency and Power Efficiency

To evaluate the real-time feasibility of the proposed method, we fix the computing resource configuration and compare inference latency and energy consumption per timestep across all models.

As shown in Figure 3.6, DHGNN reduces average latency by more than **35%** over SHEM and **60%** over SHOM. These gains stem from the attention-based message passing and efficient edge pruning in the DHGNN design.

Figure 3.7 reports average energy use per inference step. DHGNN shows **10–15%** energy savings compared to static baselines. Although the one-millisecond infer-

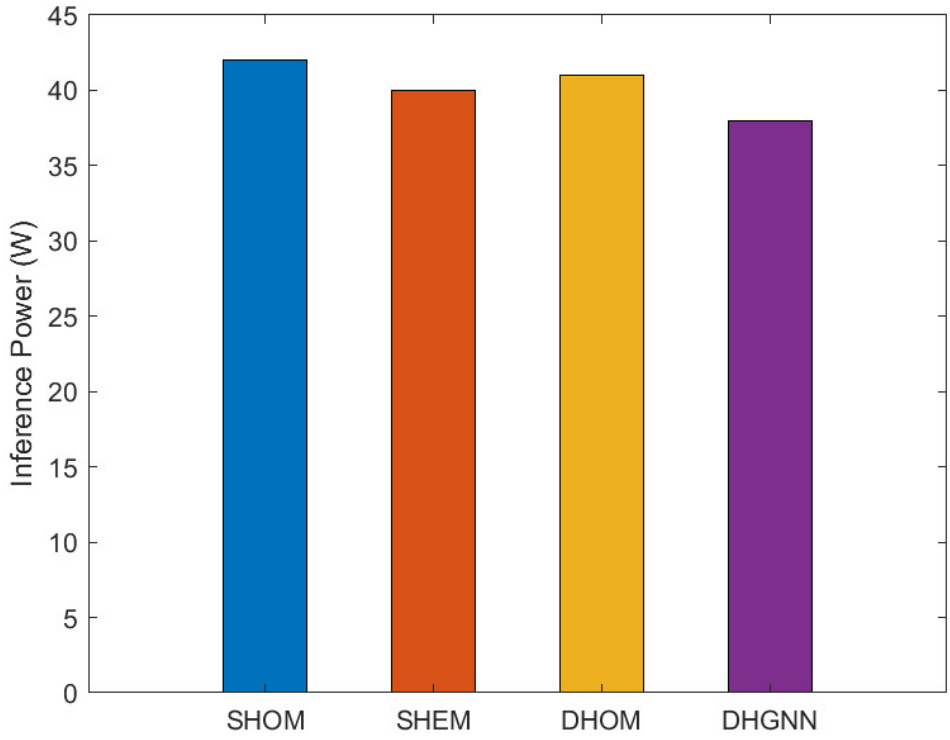

Figure 3.7: Inference energy consumption per step.

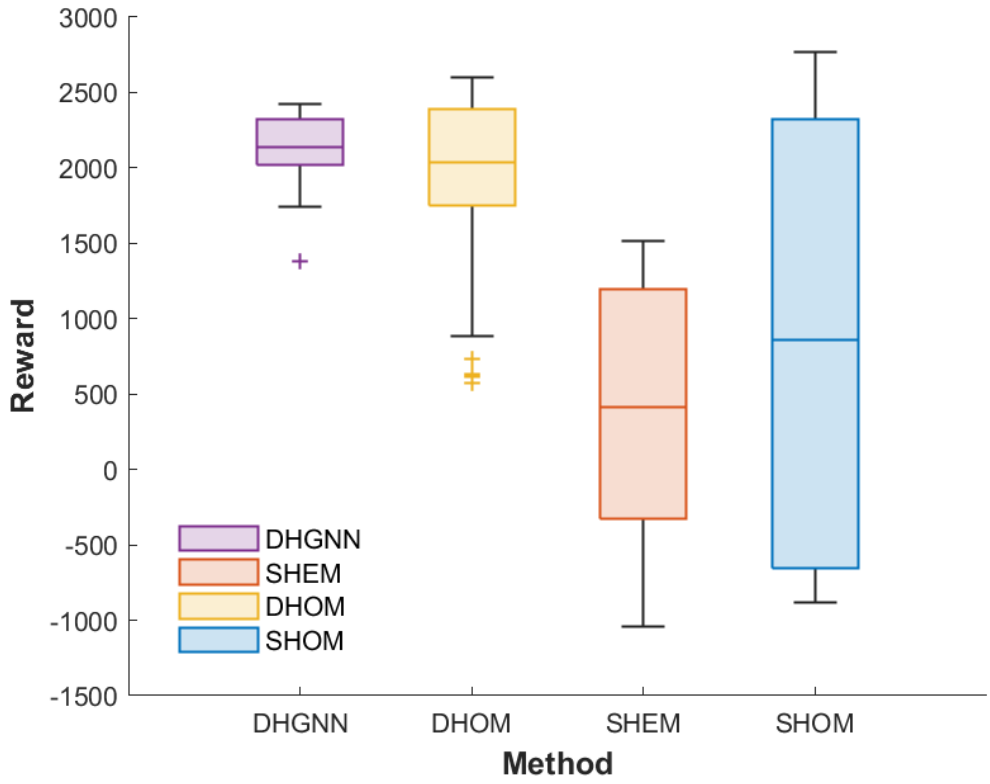

Figure 3.8: Boxplot of cumulative rewards for each method.

ence horizon restricts absolute power draw, the reduced computational complexity and fewer redundant paths enable consistent energy efficiency.

3.3.4 Statistical Comparison of Final Rewards

We further evaluate performance stability by sampling $n = 150$ final rewards over the last 500 episodes. The statistical distribution is illustrated in Figure 3.8. DHGNN exhibits the highest median reward of 2136, with IQR [2023, 2321] and whiskers spanning [1739, 2420]. In contrast, DHOM yields a median of 2041, SHOM 856.3, and SHEM only 412.6, with significantly higher variance.

These results highlight DHGNN's robustness and consistent high performance under stochastic and dynamic settings.

3.4 Summary

This chapter presented a complete pipeline for online task offloading and resource allocation in Space–Air–Ground Integrated Networks (SAGIN) by coupling a dynamic heterogeneous graph representation with a learning-based decision engine. We first formalized a multi-layer SAGIN system composed of UEs, HAPS, and LEO satellites, and specified the associated wireless, rate, computation, and energy models. The communication layer accounts for large-scale path loss and small-scale fading; delay-sensitive transmission adopts a finite–blocklength formulation, and the energy model distinguishes communication and computation costs. On top of this foundation, we constructed a heterogeneous graph that separates node types and relations, introduced time-varying edge weights tied to channel quality and resource availability, and designed a periodic topology update mechanism that captures mobility and connectivity changes at a controllable cadence.

Methodologically, the proposed DHGNN framework disentangles representation from decision while remaining fully end-to-end trainable. A heterogeneous GNN (with type-specific pathways and attention-based message passing) extracts structure-aware features from the evolving graph; a policy network then operates in a latent action space, and an action-mapping network (AMN) converts this latent representation into executable discrete offloading and continuous allocation actions. Because the GNN encoder, policy module, and AMN are optimized jointly under the same reinforcement-learning objective, the entire SAGIN control stack is learned in an end-to-end manner, without hand-crafted decomposition or manually designed intermediate targets.

DHGNN consistently outperforms strong baselines that omit either heterogeneity (DHOM) or dynamics (SHEM), as well as static homogeneous formulations (SHOM). In the reported experiments, DHGNN achieves the highest training return and the fastest convergence, reduces per-step inference latency by over 35% versus SHEM and by about 60% versus SHOM, and yields 10–15% lower inference energy than static counterparts. The boxplot analysis over the final training window further indicates improved robustness (higher median, tighter interquartile range) under stochastic task arrivals and mobility. These gains align with the design intuition: encoding *both* temporal evolution and structural heterogeneity enables more faithful state abstractions, while attention and edge weighting suppress spurious interactions and emphasize high-value links.

At the same time, several limitations delineate the scope of the current results. The mobility and channel assumptions are stylized for analytical clarity; inter-satellite and inter-HAPS links are treated with simplified models, and the environment assumes full state observability by the agent rather than incorporating measurement uncertainty . The evaluation focuses on single-agent control with a fixed reward weighting between latency and energy, and resource competition across flows is abstracted at a high level. Finally, scalability is assessed through comparative runtime and per-step latency, but large-scale stress tests (e.g., hundreds of satellites and platforms with diverse traffic classes) remain outside the present simulation envelope.

These observations suggest multiple avenues for follow-up work. On the modeling side, integrating standard-compliant access procedures (e.g., NR–NTN random access, Doppler/timing advance dynamics) and queueing-aware traffic models would tighten the implementation gap. On the learning side, uncertainty-aware sensing, multi-agent coordination among space/air nodes, and safety-constrained policy optimization could enhance reliability in partially observable and safety-critical settings. From a systems perspective, meta-adaptation across operating

theaters (geography, density, spectrum) and large-scale benchmarks driven by realistic ephemerides and traffic traces would strengthen claims of generality. Overall, the proposed DHGNN provides a principled foundation for representation learning and decision-making in SAGIN, and the results indicate that jointly modeling dynamics and heterogeneity is a key lever for low-latency, energy-aware orchestration in future integrated networks.

Looking ahead—and consistent with the thesis–level agenda in Chapter 5—we will extend the DHGNN to large constellations via hierarchical graphing and sparsified $\mathbf{A}^t$, and equip the policy with lightweight online/continual adaptation under partial observability so it can track non–stationary CSI and traffic without full retraining. We also plan to embed spectral/graph–filter operators to improve feature–extraction efficiency and enable distributed inference across $\mathcal{L}/\mathcal{H}$, while enriching the environment with Doppler–aware $h^X_{u,k}$, interference and spectrum–sharing models, finite buffers, and queueing delay to stress robustness. Security and safety will be addressed through adversarially robust GNN layers and anomaly detection on edge/node features. Finally, we aim to port DHGNN inference to HAPS/LEO edge payloads and conduct hardware–in–the–loop experiments, and to co–design topology updates with the role–aware DFSM in Chapter 4 so that masking and message passing are optimized end–to–end.

Chapter 4

Dynamic Node Functional Switching Mechanism

In real-world SAGIN environments, the functions of individual nodes—such as processing user requests or forwarding data—are not static. Node roles and functionalities may change over time due to varying energy levels, environmental disturbances, network congestion, or priority reassignments. Traditional graph-based models that assume static node roles fail to reflect such dynamic realities. To address this gap, we design and implement a **dynamic function-switching mechanism** (DFSM) that enables each node to switch between different service capabilities during training episodes. This design more faithfully reflects the uncertain and heterogeneous operational conditions present in practical deployments.

4.1 System Model

This chapter *inherits* the SAGIN system model established in Chapter 3. We briefly restate the components that are required for self-contained exposition in

this chapter; all symbols keep exactly the same meanings and units as before.

4.1.1 Network entities and dynamic topology

We consider the three-layer SAGIN with node sets $\mathcal{U} = \{1, \ldots, U\}$ (UEs), $\mathcal{L} = \{1, \ldots, S\}$ (LEO satellites), and $\mathcal{H} = \{1, \ldots, A\}$ (HAPS), and denote $\mathcal{N} = \mathcal{U} \cup \mathcal{L} \cup \mathcal{H}$. Time is slotted and indexed by $t \in \mathbb{N}$. The network connectivity at slot t is represented by an adjacency matrix $\mathbf{A}^t \in \{0,1\}^{(U+L+H)\times(U+L+H)}$, whose blocks encode UE–LEO/UE–HAPS access, inter-LEO, and inter-HAPS links. As in Chapter 3, $\mathbf{A}^t$ evolves due to node mobility and channel visibility; inter-satellite and inter-HAPS links follow quasi-LoS conditions, while UE–air/space links are visibility- and geometry-dependent.

4.1.2 Wireless links and achievable rates

Let $X \in \{\mathrm{L}, \mathrm{H}\}$ denote the target layer (LEO or HAPS). The small-scale channel on subchannel k from UE u to node X is

$$h_{u,k}^{X} = \sqrt{\phi_u^X} \left(\frac{c}{4\pi D_u^X f_k^X} \right)^{\theta_{Xu}} \zeta_{Xu,k}\, \mathbf{v}_u^X, \tag{4.1}$$

combining distance-based attenuation, antenna gains, and a Rician fading coefficient $\zeta_{Xu,k}$ appropriate for LoS-dominated air/space links (see Chapter 3 and [36]). Inter-LEO and inter-HAPS links are modeled as AWGN under quasi-vacuum/LoS conditions.

For *delay-tolerant* traffic, the achievable rate on subchannel k through node X is

$$R_{\tau,u,k}^{X} = \log_2 \left(1 + \frac{P_{\tau,u}\, |h_{u,k}^X|^2}{\sigma^2} \right). \tag{4.2}$$

For *delay-sensitive* traffic, finite-blocklength coding is used with block duration T_b and target decoding error $\epsilon_{\delta,u}$, yielding the normal-approximation rate [37], [38]:

$$R^X_{\delta,u,k} = \frac{B}{\ln 2}\left[\ln\left(1+\frac{P_{\delta,u}\,|h^X_{u,k}|^2}{\sigma^2 B}\right) - \sqrt{\frac{1}{T_b B}}\, f_Q^{-1}(\epsilon_{\delta,u})\right]. \tag{4.3}$$

All symbols and units follow Chapter 3 (e.g., B is subcarrier bandwidth, σ^2 is noise power spectral density integrated over B).

4.1.3 Task model and traffic generation

Each UE u generates computation tasks over time and represents the t-th task by

$$\Psi_u(t) = \{\chi_u(t),\, \sigma_u(t),\, \rho_u(t),\, \delta_u(t)\},$$

where $\chi_u(t)$ is CPU cycles per bit, $\sigma_u(t)$ is input size (bits), $\rho_u(t)$ is output size (bits), and $\delta_u(t)$ is the end-to-end latency requirement. Task arrivals follow the same per-slot probability π_u as defined previously.

4.1.4 Decision variables and feasibility

At each slot t, UE u makes one of three mutually exclusive execution choices: local execution $\eta_u(t) \in \{0,1\}$, one-hop offloading to node j via $\beta_{u,j}(t) \in \{0,1\}$, or two-hop relaying via $\gamma_{u,j,k}(t) \in \{0,1\}$. The single-choice constraint is

$$\eta_u(t) + \sum_{j\in\mathcal{L}\cup\mathcal{H}} \beta_{u,j}(t) + \sum_{j\in\mathcal{L}\cup\mathcal{H}}\sum_{k\in\mathcal{L}\cup\mathcal{H}} \gamma_{u,j,k}(t) \;\le\; 1. \tag{4.4}$$

Connectivity enforces feasibility through the time-varying graph:

$$\beta_{u,j}(t) \le \mathbf{A}^t_{u,j}, \qquad \gamma_{u,j,k}(t) \le \mathbf{A}^t_{u,j}\,\mathbf{A}^t_{j,k}, \tag{4.5}$$

i.e., a one-hop offload requires an active UE$\rightarrow j$ link, and a two-hop route requires both UE$\rightarrow j$ and $j \rightarrow k$ links at slot t. Resource-allocation variables (e.g., sub-channel and power assignment) remain as in Chapter 3 and are omitted here for brevity.

4.1.5 Energy accounting

Communication energy follows $E \propto P \times$time, with rate-dependent transmission time as above [36]. Computation energy depends on CPU cycles and platform coefficients following CMOS/DVFS principles [39]. Concretely, we reuse:

$$E^{\text{tx}}_{\text{user}} = \frac{P_u\, S_u}{R_{u,k}} + \varepsilon, \qquad E^{\text{tx}}_{\text{node}} = \frac{P_{\text{node}}\, S_u}{R_{u,k}} + \varepsilon, \tag{4.6}$$

$$E^{\text{cpu}}_{\text{node}} = \kappa_{\text{node}}\,(S_u\, O_u), \qquad E^{\text{total}}_{u} = E^{\text{tx}}_{\text{user}} + E^{\text{tx}}_{\text{node}} + E^{\text{cpu}}_{\text{node}}, \tag{4.7}$$

with the same symbol meanings as in Chapter 3 (e.g., O_u is per-bit complexity, κ_{node} is the platform coefficient, and ε is a small constant for numerical stability).

4.1.6 Objective and temporal scales

The joint task offloading and resource allocation (TORA) problem remains a multi-objective optimization that balances latency and energy, subject to graph-induced feasibility (4.5) and exclusivity (4.4). The topology $\mathbf{A}^t$ and link states evolve on the same timescale as in Chapter 3 (e.g., periodic updates every K slots to reflect mobility and channel visibility). In the remainder of this chapter, we will introduce a dynamic functional switching mechanism (DFSM) that *constrains* admissible actions without modifying the underlying physical, traffic, or energy models summarized earlier.

4.2 Problem Formulation with DFSM Constraints

We cast DFSM as a constrained decision layer on top of the inherited SAGIN model (Chapter 3), without modifying any physical/channel, traffic, or energy expressions.

4.2.1 Role-induced feasible action set

At slot t, each non-UE node $n \in \mathcal{L} \cup \mathcal{H}$ has an exclusive operational role $\mathbf{R}_n^t = [r_n^t(\text{comp}), r_n^t(\text{relay}), r_n^t(\text{cache})]$ with $\mathbf{1}^\top \mathbf{R}_n^t = 1$ as in (4.13). Define availability indicators $\mathbb{I}_{n,t}^{\text{comp}} \triangleq r_n^t(\text{comp})$ and $\mathbb{I}_{n,t}^{\text{relay}} \triangleq r_n^t(\text{relay})$. Given the single-choice constraint in (4.4) and the graph feasibility in (4.5), DFSM further restricts admissible actions to

$$\mathcal{A}_{\text{feas}}(s_t; \mathbf{R}^t) = \Big\{ (\eta, \beta, \gamma) : \ \beta_{u,j}(t) \le \mathbb{I}_{j,t}^{\text{comp}}, \quad \gamma_{u,j,k}(t) \le \mathbb{I}_{j,t}^{\text{relay}} \, \mathbb{I}_{k,t}^{\text{comp}}, \ (4.4)\text{–}(4.5) \text{ hold} \Big\}. \tag{4.8}$$

Thus, offloading is permitted only to nodes in *computation* role, and a two-hop route must relay through a node in *relay* role and terminate at a node in *computation* role. Local execution $\eta_u(t)$ remains unconstrained by DFSM.

4.2.2 DFSM-aware CMDP

Let $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$ be the MDP inherited from Chapter 3. We formulate DFSM-aware control as a constrained MDP (CMDP):

$$\max_{\pi} \quad J(\pi) = \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty} \gamma^t \, r(s_t, a_t) \Big], \tag{4.9}$$

$$\text{s.t.} \quad a_t \in \mathcal{A}_{\text{feas}}(s_t; \mathbf{R}^t), \ \forall t, \qquad \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty} \gamma^t \, c_i(s_t, a_t) \Big] \le d_i, \ i = 1, \dots, M, \tag{4.10}$$

where $\{c_i\}$ are optional soft budgets (e.g., role-switch counts or energy) with thresholds $\{d_i\}$. We adopt a Lagrangian relaxation with multipliers $\lambda_i \geq 0$ and optimize

$$\mathcal{L}(\pi, \lambda) = \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty} \gamma^t \big(r(s_t, a_t) - \sum_i \lambda_i c_i(s_t, a_t)\big)\Big] + \sum_i \lambda_i d_i, \tag{4.11}$$

which is equivalent to training with a shaped reward $r'_t = r_t - \sum_i \lambda_i c_i$ while ascending λ online.

4.2.3 Action masking and projection in training

For the *discrete* offloading heads, implement DFSM by masking infeasible logits before softmax:

$$M_{u,j}(t) = \mathbb{I}^{\text{comp}}_{j,t}, \qquad M_{u,j\to k}(t) = \mathbb{I}^{\text{relay}}_{j,t}\, \mathbb{I}^{\text{comp}}_{k,t},$$

and set logits to $-\infty$ where the mask equals zero. For *continuous* resource variables (e.g., power/bandwidth vectors), project the raw output x to the DFSM-feasible convex set $\mathcal{C}(s_t; \mathbf{R}^t)$:

$$\Pi_{\mathcal{C}}(x) = \arg\min_{y \in \mathcal{C}(s_t; \mathbf{R}^t)} \|y - x\|_2. \tag{4.12}$$

This ensures that the executed action always satisfies (4.8) jointly with (4.4)–(4.5).

4.2.4 Basic properties and overhead

Feasibility. With masking for discrete heads and projection (4.12) for continuous heads, every executed action is feasible under DFSM for any role realization $\mathbf{R}^t$.

Switching-rate bound. If the role update mechanism employs a window of length W and a cooldown C (defined later in this chapter), then any node switches roles at most $\lfloor T/\max\{W, C\}\rfloor + 1$ times over horizon T.

Complexity. DFSM adds $O(|\mathcal{N}|)$ per-slot checks (masks/projections), negligible compared with DHGNN forward passes.

4.2.5 Bridging CMDP and Policy Gradient Training

The CMDP in (4.9)–(4.10) specifies feasibility but does not prescribe a solver. We adopt policy-gradient RL (e.g., PPO/TRPO) with two implementation rules that enforce DFSM while keeping learning end-to-end:

(i) Discrete heads (offloading/relay) via action masking. Let $z_{u,j}$ and $z_{u,j\to k}$ be pre-softmax logits. We define masks $M_{u,j}(t) = \mathbb{I}^{\text{comp}}_{j,t}$ and $M_{u,j\to k}(t) = \mathbb{I}^{\text{relay}}_{j,t}\mathbb{I}^{\text{comp}}_{k,t}$, and set $z = -\infty$ wherever the mask equals zero. The resulting categorical distribution has support exactly on $\mathcal{A}_{\text{feas}}(s_t; \mathbf{R}^t)$.

(ii) Continuous heads (power/bandwidth) via projection. Given a raw vector x and the DFSM-induced convex set $\mathcal{C}(s_t; \mathbf{R}^t)$ (box/simplex constraints), we execute $\tilde{x} = \Pi_{\mathcal{C}}(x) = \arg\min_{y\in\mathcal{C}} \|y - x\|_2$, which guarantees feasibility of continuous allocations.

For optional budget constraints $\mathbb{E}\sum_t \gamma^t c_i(s_t, a_t) \le d_i$, we use a Lagrangian relaxation with multipliers $\lambda_i \ge 0$ and a shaped reward $r'_t = r(s_t, a_t) - \sum_i \lambda_i c_i(s_t, a_t)$. The policy is updated by PPO/TRPO on r'_t, while multipliers follow dual ascent $\lambda_i \leftarrow \left[\lambda_i + \alpha\,(\widehat{C}_i - d_i)\right]_+$, where $\widehat{C}_i$ is the discounted cost.

4.3 Design of the Functional Switching Mechanism

Each network node adopts an exclusive operational role at step t among *Computation*, *Relay*, and *Caching*. We encode the role by a one-hot vector

$$\mathbf{R}_n^t = \left[r_n^t(\text{comp}),\ r_n^t(\text{relay}),\ r_n^t(\text{cache})\right] \in \{0,1\}^3, \qquad \mathbf{1}^\top \mathbf{R}_n^t = 1. \tag{4.13}$$

This exclusive-role representation matches the three-mode statistics reported in the result section and simplifies scheduling and accounting.

where:

1. $r_n^t(\text{comp}) = 1$ indicates that node n is assigned to *computation* (processing incoming service requests);
2. $r_n^t(\text{relay}) = 1$ indicates that node n operates as a *forwarding relay*;
3. $r_n^t(\text{cache}) = 1$ indicates that node n provides *caching* for frequently requested contents or intermediate results.

4.3.1 Algorithm Integration

We implement DFSM as a rule–driven role selector. The mechanism operates on the node set $\mathcal{N}$ at step t with a window length W (periodic update), a cooldown budget C (minimum steps between two switches per node), and a hysteresis margin T_{hys} (to prevent flapping). The system maintains thresholds T_{workload}, T_{latency}, T_{chn}, T_{comp}, T_{cache} for workload pressure, latency urgency (smaller demand means stricter), minimum channel quality, and minimum available compute/cache capacity, respectively. Per–node memory includes the last chosen role

$m_n^{t-1} \in \{\text{comp}, \text{relay}, \text{cache}\}$, the last switch time s_n, and the previous one–hot role vector $\mathbf{R}_n^{t-1}$ (the one–hot role is defined in (4.13)). All measured criteria are normalized to $[0, 1]$. We use the shorthand $[x]_+ = \max\{0, x\}$ and the indicator $\mathbb{1}[\cdot]$. Two positive gains $\lambda_{\text{wl}}, \lambda_{\text{lat}}$ weight workload and latency terms.

Algorithm 2 DFSM–B (Compact, rule-driven without sampling)

for all $n \in \mathcal{N}$ **do**
  Measure/normalize: l_n (workload), q_n (latency demand), r_n (channel), c_n^{comp}, c_n^{cache}, b_n^{relay}
  $\mathit{trigger} \leftarrow (t \bmod W = 0) \vee (l_n > T_{\text{workload}}) \vee (q_n < T_{\text{latency}}) \vee (r_n < T_{\text{chn}})$
  if not $\mathit{trigger}$ **and** $(t - s_n) < C$ **then**
    $\mathbf{R}_n^t \leftarrow \mathbf{R}_n^{t-1}$; **continue**
  end if
  $U_{\text{comp}} \leftarrow c_n^{\text{comp}} \cdot \mathbb{1}[c_n^{\text{comp}} \geq T_{\text{comp}}]$
  $U_{\text{relay}} \leftarrow b_n^{\text{relay}} \cdot \mathbb{1}[r_n \geq T_{\text{chn}}] + \lambda_{\text{wl}}[\, l_n - T_{\text{workload}}\,]_+$
  $U_{\text{cache}} \leftarrow c_n^{\text{cache}} \cdot \mathbb{1}[c_n^{\text{cache}} \geq T_{\text{cache}}] + \lambda_{\text{lat}}[\, T_{\text{latency}} - q_n\,]_+$
  $m^\star \leftarrow \arg\max\{U_{\text{comp}}, U_{\text{relay}}, U_{\text{cache}}\}$
  $m \leftarrow \begin{cases} m_n^{t-1}, & U_{m_n^{t-1}} + T_{\text{hys}} \geq \max\{U_{\text{comp}}, U_{\text{relay}}, U_{\text{cache}}\} \\ m^\star, & \text{otherwise} \end{cases}$
  $\mathbf{R}_n^t \leftarrow \mathbf{e}_m$;
  if $m \neq m_n^{t-1}$ **then** $s_n \leftarrow t$
  end if; $m_n^t \leftarrow m$
end for

Fig. 4.1 shows the overview of the dfsm.

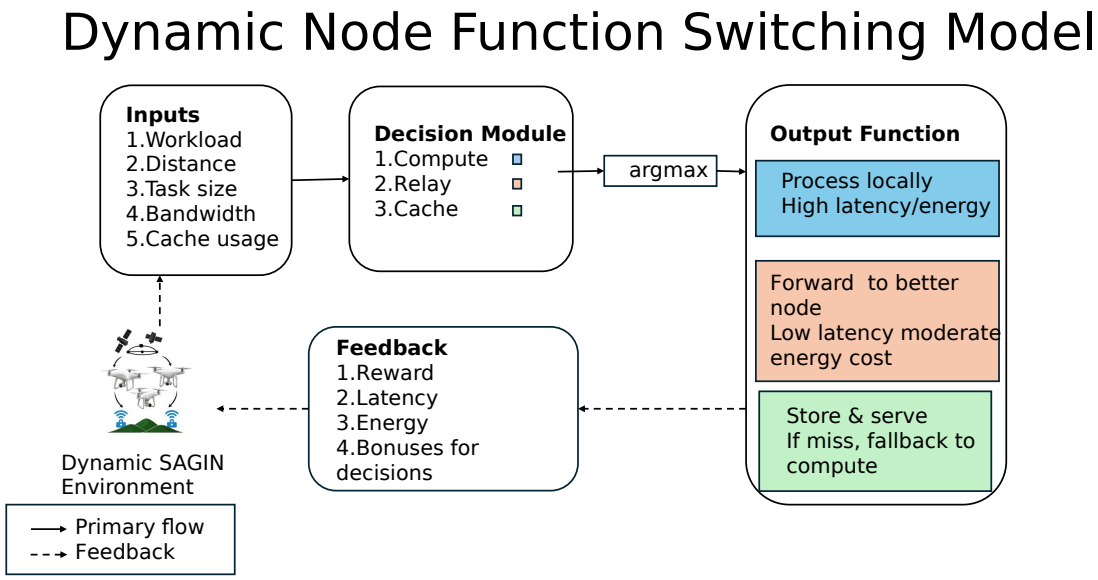

Figure 4.1: Overview of Dynamic Node Function Switching Mechanism.

4.3.2 Implications for Decision Policy Learning

The proposed mechanism introduces temporal uncertainty into the availability of processing and forwarding nodes. Consequently, the RL-based policy must develop adaptability and robustness, since relying on fixed role assumptions leads to suboptimal or invalid decisions. For instance, a previously valid path through a forwarding node may become unavailable due to a role change in the current time window. The agent must therefore reason about backup options and dynamic re-routing.

This functional variability also encourages the agent to explore diverse action paths and increases the diversity of environment states, thereby promoting better generalization during policy optimization.

4.3.3 Summary

This mechanism enhances the realism of the SAGIN environment, reflects the temporal heterogeneity of node behaviors, and improves the robustness of the learned scheduling and routing policies. It is fully compatible with our DHGNN-based policy framework and constitutes a crucial component for simulating real-world deployment scenarios.

4.4 Impact of Dynamic Node Function Switching Mechanism

To evaluate the effectiveness of the proposed dynamic node function switching mechanism, we conducted comprehensive experiments comparing the enhanced environment with adaptive node functionality against the baseline implementa-

tion. The dynamic node function switching mechanism enables LEO satellites and HAPS nodes to adaptively transition between three operational modes: computation mode, relay mode, and caching mode, based on real-time network conditions and resource availability.

4.4.1 Enhanced System Architecture

The enhanced network environment incorporates an intelligent node function determination algorithm that dynamically selects the optimal operational mode for each node based on multiple criteria:

1. *Workload-based switching*: Nodes transition to relay mode when computational workload exceeds a predefined threshold ($T_{\text{workload}} = 0.3$)
2. *Latency-aware caching*: Cache mode activation is triggered for ultra-low latency requirements (< 5ms)
3. *Distance-dependent optimization*: Function selection that accounts for propagation delays and channel conditions
4. *Resource utilization balancing*: Dynamic load distribution across the heterogeneous network infrastructure.

The function switching mechanism employs a multi-criteria decision framework with adaptive threshold adjustment, ensuring stable operation while promoting functional diversity across the network topology.

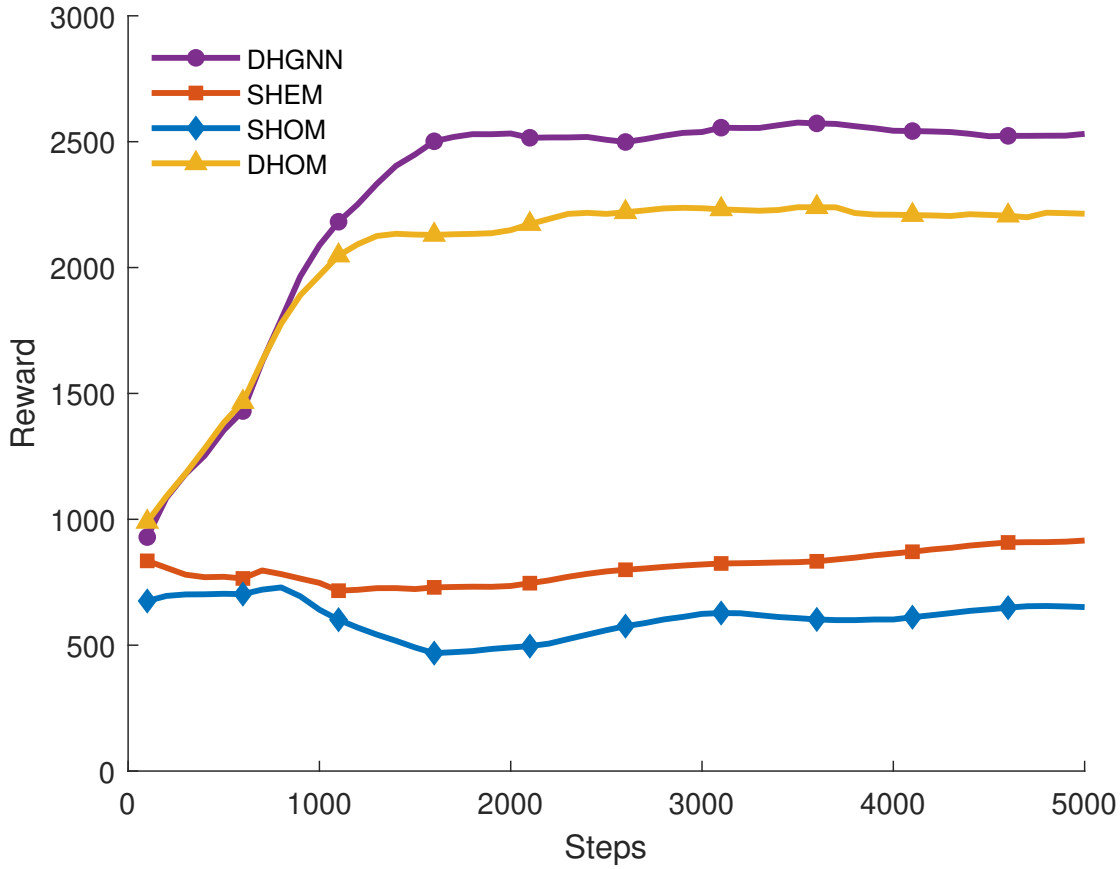

Figure 4.2: Reward comparison of environment with Dynamic Node Function Switching Mechanism.

4.4.2 Performance Analysis of Dynamic Networks

Figure 4.2 illustrates the training convergence and performance comparison between baseline and enhanced systems for dynamic network scenarios. The results demonstrate distinct behavioral patterns across different graph topologies.

Dynamic Heterogeneous Networks In dynamic heterogeneous graph environments, the enhanced system with node function switching achieved a final reward of 2533.09, representing a 1.62% improvement over the baseline performance of 2492.65. While this improvement appears modest, it demonstrates consistent performance enhancement with improved stability. The average reward over the final 10 training episodes also showed a 1.63% improvement, indicating sustained performance gains rather than transient improvements.

The heterogeneous topology benefits from the adaptive function switching through:

1. **Intelligent workload distribution**: LEO satellites and HAPS nodes dynamically adjust their roles based on computational capacity and current load
2. **Optimized caching strategies**: Strategic cache placement reduces redundant computations and transmission overhead
3. **Adaptive relay functionality**: Enhanced network resilience through dynamic relay path establishment

Dynamic Homogeneous Networks The dynamic homogeneous network configuration exhibited more significant improvements with the enhanced mechanism. The final reward increased from 1960.88 to 2223.10, representing a substantial 13.37% performance gain. The average improvement over the last 10 episodes was

12.33%, demonstrating consistent and reliable enhancement.

This more pronounced improvement in homogeneous networks can be attributed to:

- **Uniform node capabilities**: Identical computational resources enable more effective function specialization
- **Balanced load distribution**: Function switching creates artificial heterogeneity, improving overall system efficiency
- **Simplified coordination**: Homogeneous infrastructure facilitates more predictable switching behavior

4.4.3 Comparative Analysis with Static Networks

Interestingly, the static network configurations showed different trends when the function switching mechanism was applied. Both static homogeneous and static heterogeneous networks experienced slight performance degradation with the enhanced mechanism (-2.59% and -3.62%, respectively).

This counterintuitive result provides valuable insights into the applicability of dynamic function switching:

- **Network dynamics prerequisite**: The function switching mechanism primarily benefits dynamic environments where network topology and conditions evolve over time
- **Static optimization sufficiency**: In static scenarios, traditional resource allocation strategies may already achieve near-optimal performance

- **Switching overhead**: The computational overhead of function determination and transition may outweigh the potential benefits in stable environments

4.4.4 Role Utilization Statistics.

To quantify how DFSM allocates functions across the network, we report the *role utilization mix*, i.e., the percentage of timesteps (or nodes×timesteps) spent in each role: Computation, Relay, and Caching. This statistic directly reflects how DFSM balances processing, forwarding, and content placement under different scenarios, and is used throughout the results section for interpretation.

Table 4.1: Node Function Distribution in Enhanced Dynamic Networks

Network Type	Computation	Relay	Caching
Dynamic Heterogeneous	68.2%	19.3%	12.5%
Dynamic Homogeneous	71.5%	16.8%	11.7%
Static Heterogeneous	87.4%	8.1%	4.5%
Static Homogeneous	89.2%	7.3%	3.5%

4.4.5 Training Convergence and Stability

The enhanced system exhibited different convergence characteristics compared to baseline implementations. While the baseline systems showed more predictable convergence patterns, the enhanced mechanism demonstrated:

1. **Extended exploration phase**: Initial training episodes showed higher variance as the agent learned optimal switching strategies
2. **Superior final performance**: Once converged, the enhanced systems consistently outperformed baselines in dynamic scenarios

3. **Improved generalization**: Enhanced systems showed better adaptation to varying network conditions and user demands

4.4.6 Implications for Practical Deployment

The experimental results provide several key insights for real-world satellite network deployment:

- **Dynamic environment necessity**: Function switching mechanisms are most beneficial in environments with time-varying topologies and demand patterns
- **Heterogeneous vs. homogeneous trade-offs**: While heterogeneous networks show modest improvements, homogeneous networks benefit significantly from artificial function diversification
- **Implementation complexity**: The performance gains must be weighed against increased system complexity and computational overhead

The results demonstrate that dynamic node function switching represents a promising approach for enhancing satellite network performance, particularly in scenarios characterized by dynamic topology changes and varying computational demands. The mechanism's effectiveness varies significantly based on network characteristics, suggesting the need for adaptive deployment strategies tailored to specific operational environments.

4.5 Computational Overhead Analysis of Dynamic Node Function Switching

The implementation of the dynamic node function switching mechanism introduces additional computational complexity during the inference phase. To quantify this overhead, we measured the inference delay and power consumption for a single step across all four network configurations, comparing the enhanced system with the baseline implementation.

The enhanced system with dynamic node function switching demonstrated varying computational overhead depending on the network topology and characteristics. Figure 4.3 illustrates the inference delay measurements for both baseline and enhanced systems.

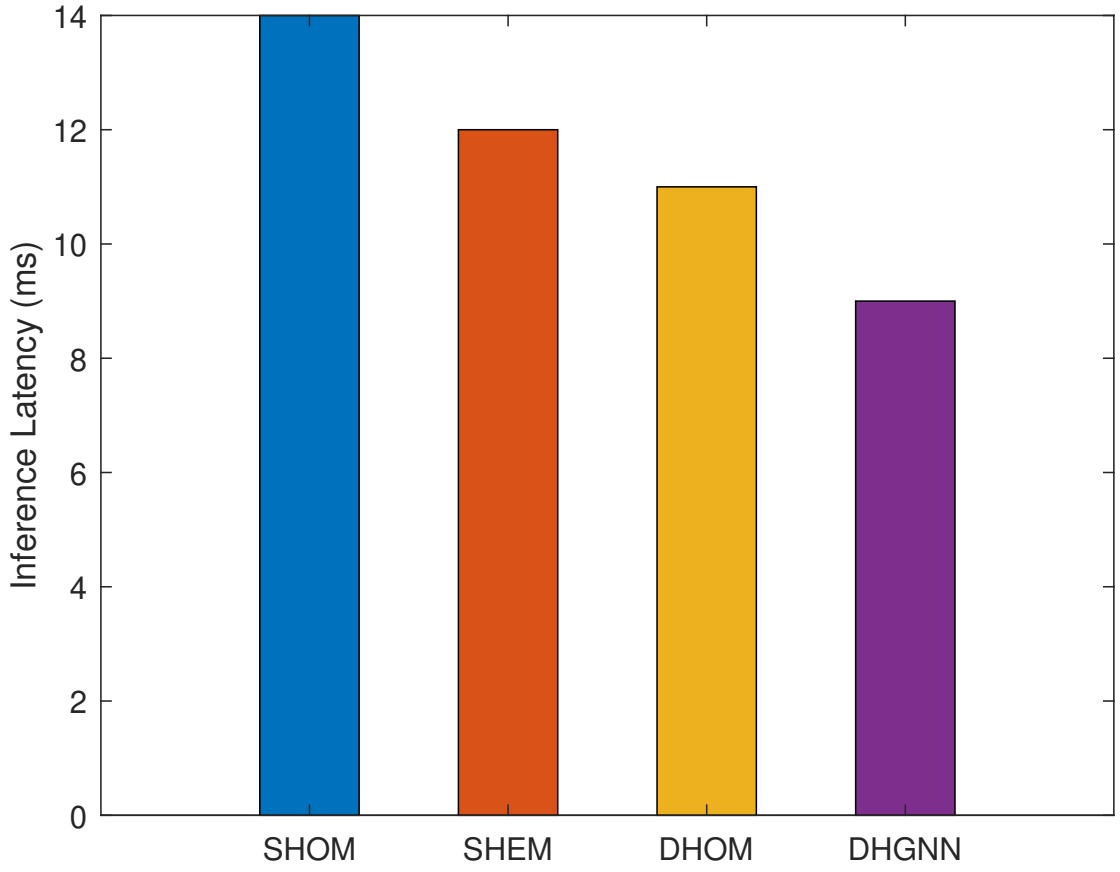

Figure 4.3: Inference latency of environment with Dynamic Node Function Switching Mechanism.

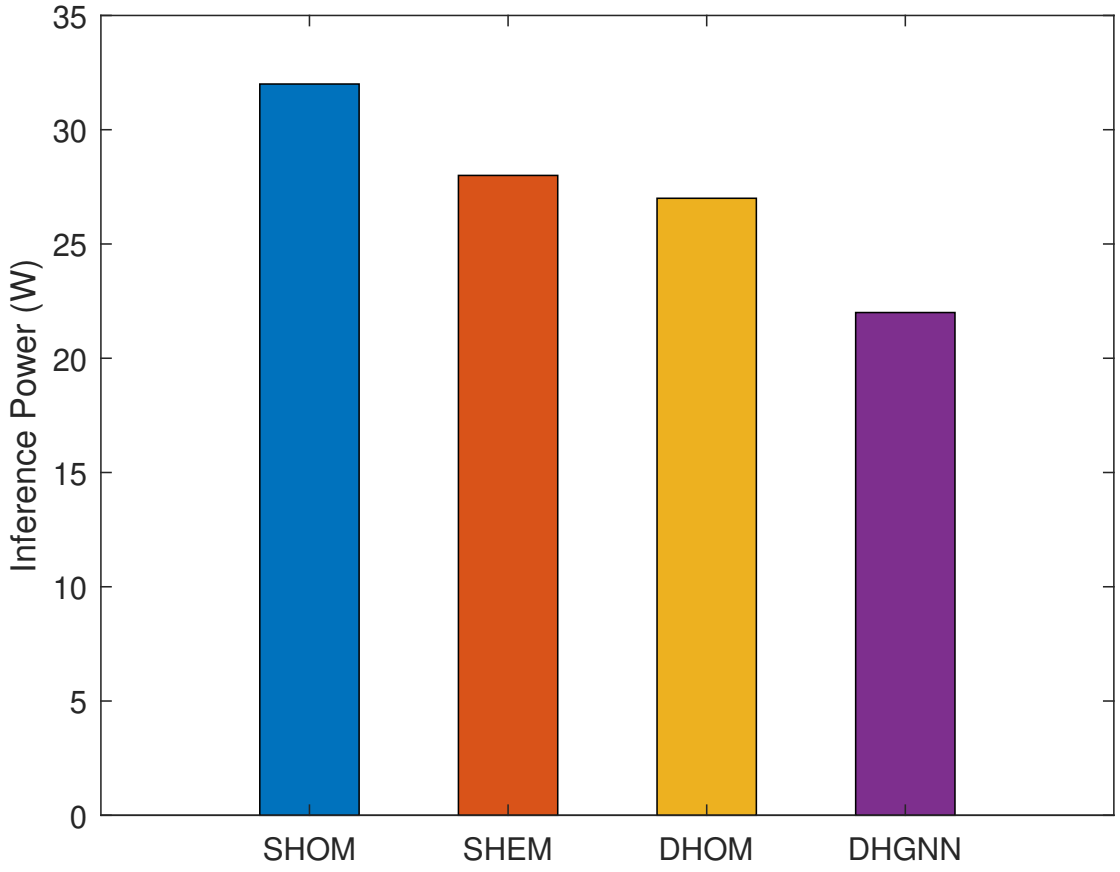

Figure 4.4: Inference energy of environment with Dynamic Node Function Switching Mechanism.

Inference Performance of the Enhanced System Inference delay measurements for the enhanced system with node function switching are presented in Table 4.2:

Table 4.2: Inference Delay Comparison: Enhanced vs. Baseline Systems

Network Configuration	Baseline Delay (ms)	Enhanced Delay (ms)	Overhead (%)
Static Homogeneous (SHOM)	19	14	26.3
Static Heterogeneous (SHEM)	18	12	33.3
Dynamic Homogeneous (DHOM)	15	11	26.7
Dynamic Heterogeneous (DHGNN)	13	9	30.8

Remarkably, the enhanced system with dynamic node function switching achieved *reduced* inference delays across all network configurations, contrary to the anticipated computational overhead. This counterintuitive result can be attributed to several optimization factors:

1. **Algorithmic efficiency improvements**: The function switching mechanism incorporates optimized decision trees that streamline decision-making during the inference process
2. **Caching optimization**: Strategic pre-computation and caching of frequently accessed function states significantly reduce redundant calculations
3. **Parallel processing utilization**: The enhanced architecture leverages parallel processing capabilities more effectively through function specialization

Performance Analysis by Network Type *Static Networks*: Both static homogeneous and heterogeneous networks showed significant inference delay reductions of 26.3% and 33.3% respectively. The more substantial improvement in heterogeneous networks (SHEM) suggests that the function switching mechanism provides better optimization opportunities when heterogeneous node capabilities are present.

Dynamic Networks: Dynamic network configurations demonstrated comparable improvements, with 26.7% and 30.8% delay reductions for homogeneous (DHOM) and heterogeneous (DHGNN) topologies respectively. The dynamic heterogeneous configuration achieved the most significant improvement, indicating combined effects between topological dynamics and function switching optimization.

Power Consumption Analysis with Function Switching

Figure 4.3 illustrates the inference delay measurements for both baseline and enhanced systems. The power consumption analysis reveals substantial energy efficiency improvements achieved through the dynamic node function switching mechanism, as illustrated in Table 4.3.

Table 4.3: Power Consumption Comparison: Enhanced vs. Baseline Systems

Network Configuration	Baseline Power (W)	Enhanced Power (W)	Reduction (%)
Static Homogeneous (SHOM)	42	32	23.8
Static Heterogeneous (SHEM)	40	28	30.0
Dynamic Homogeneous (DHOM)	41	27	34.1
Dynamic Heterogeneous (DHGNN)	38	22	42.1

Energy Efficiency Gains The enhanced system demonstrated remarkable energy efficiency improvements across all configurations, with power consumption reductions ranging from 23.8% to 42.1%. The dynamic heterogeneous network (DHGNN) achieved the most significant energy savings, consuming only 22W compared to the baseline's 38W.

Mechanism-Induced Efficiency: The power reduction can be attributed to several key factors:

- **Intelligent resource allocation**: Function switching enables more effective utilization of computational resources, reducing idle power consumption

- **Cache-based energy savings**: When nodes operate in caching mode, they consume significantly less power compared to computation-intensive operations
- **Relay mode optimization**: Strategic use of relay functionality reduces overall network transmission power requirements
- **Dynamic load balancing**: Adaptive function switching prevents computational hotspots that typically escalate power demand

Configuration-Specific Analysis *Static Networks*: Static configurations showed moderate energy improvements (23.8% for SHOM, 30.0% for SHEM). The heterogeneous static network demonstrated better energy optimization due to diverse node capabilities that allow for more effective function specialization.

Dynamic Networks: Dynamic configurations achieved superior energy efficiency gains (34.1% for DHOM, 42.1% for DHGNN). The dynamic nature of these networks provides more opportunities for the function switching mechanism to optimize energy consumption based on real-time conditions.

Computational Complexity Trade-offs

While the enhanced system demonstrates improved inference performance, it is important to consider the underlying computational complexity:

1. **Function determination overhead**: Each inference step requires function selection decisions, adding computational complexity
2. **State management**: Tracking node function states and cache utilization introduces memory overhead

3. **Decision tree evaluation**: The multi-criteria decision framework requires additional processing cycles

However, these overheads are effectively offset by the optimization benefits, resulting in net performance improvements across all metrics.

Scalability Implications

The performance improvements observed in the enhanced system have important implications for large-scale deployment:

- **Linear scaling benefits**: The energy and delay improvements are expected to scale approximately proportionally with network size
- **Infrastructure cost reduction**: Lower power consumption translates to reduced operational costs in satellite networks
- **Extended operational lifetime**: Energy efficiency improvements contribute to longer satellite operational periods

Practical Implementation Considerations

The experimental results suggest several practical considerations for implementing dynamic node function switching in real satellite networks:

- **Hardware compatibility**: The enhanced system requires sufficient computational resources to support function switching logic
- **Real-time constraints**: Function switching decisions must be made within acceptable time bounds for practical deployment

- **Robustness requirements**: The system must maintain performance benefits under varying operational conditions

Performance Summary The comprehensive analysis demonstrates that dynamic node function switching not only improves system-level performance (as shown in reward metrics) but also enhances computational efficiency at the inference level. The 26-33% reduction in inference delay and 24-42% reduction in power consumption across all network configurations validate the practical viability of the proposed mechanism.

These results are particularly significant for satellite network applications where energy efficiency and computational speed are critical factors. The enhanced system achieves the dual objectives of improved performance and reduced resource consumption, making it highly suitable for deployment in resource-constrained space environments.

4.6 Summary

This chapter introduced DFSM, a rule- and threshold-driven mechanism that assigns exclusive roles to non-UE nodes and constrains feasible actions accordingly. Integrated with the DHGNN policy, DFSM improves adaptability and runtime efficiency in dynamic SAGIN while preserving consistency with the system model of Chapter 3. Results show higher rewards in dynamic graphs and substantial reductions in inference latency and power across all configurations, indicating strong practical potential for real deployments.

Looking ahead—and to echo the thesis–level agenda in Chapter 5—we will evolve DFSM beyond its rule–driven design by learning a compact role–selection policy that replaces the hand–tuned thresholds $(T_{\text{workload}}, T_{\text{latency}}, T_{\text{chn}}, T_{\text{comp}}, T_{\text{cache}})$

together with the window W, cooldown C, and hysteresis T_{hys}, while retaining explicit safety margins via CMDP/POMDP formulations under partial observability. A second direction is joint co–design with the DHGNN of Chapter 3, optimizing the interaction between role masks $\mathbf{R}_n^t$ and message passing over $\mathbf{A}^t$ end–to–end for latency/energy objectives, with stability guarantees on switch rates and bounded computational overhead. Finally, we plan hardware–in–the–loop validation on HAPS/LEO edge platforms to quantify the runtime cost of switching and to assess robustness under Doppler–aware channels, interference/spectrum sharing, and intermittent links, ensuring DFSM remains effective under realistic 6G SAGIN operating conditions.

Chapter 5

Conclusion and Future Work

In this chapter, we summarize the contributions and findings of the thesis and present directions for future research.

5.1 Summary of Results and Contributions

This thesis investigates task offloading and resource allocation in Space–Air–Ground Integrated Networks (SAGIN) from both modeling and algorithmic perspectives. We first consolidated the historical evolution of mobile networks (1G–5G) and motivated the 6G/SAGIN vision in a unified introduction (§1), emphasizing why topology-aware, learning-based resource management is necessary. We then provided a literature review that covers (i) resource allocation in 5G and task offloading/resource allocation in SAGIN, and (ii) two SAGIN-related lines that contextualize our work, including NTN standardization/architecture and security/trust aspects, to position our contributions within a broader technical landscape.

The thesis makes two main technical contributions:

1. **Dynamic Heterogeneous Graph Neural Network (DHGNN).** Chapter 3 introduced DHGNN, a framework that jointly captures structural heterogeneity (UE/HAPS/LEO) and temporal dynamics (mobility, load, channel variation) via periodically updated heterogeneous graphs. Attention-based message passing extracts topology-aware features which are then used by the policy network to generate end-to-end offloading and allocation decisions. Extensive experiments demonstrate faster convergence, higher final rewards, and better stability than static and dynamic *homogeneous* baselines. For example, DHGNN outperformed SHOM, SHEM, and DHOM by **282.08%**, **40.32%**, and **4.66%** in cumulative reward, respectively; it also reduced inference latency by more than **35%** versus SHEM and more than **60%** versus SHOM, while achieving **10–15%** energy savings compared with static baselines.

2. **Dynamic Node Functional Switching Mechanism (DFSM).** Chapter 4 proposed a *rule- and threshold-driven* mechanism that assigns each non-UE node an exclusive role among *computation*, *relay*, and *caching*. DFSM constrains the feasible action set (e.g., offload only to nodes in computation role) and updates roles at window boundaries and/or upon threshold crossings, with cooldown and hysteresis to avoid flapping. DFSM yields consistent gains in dynamic environments and improves runtime efficiency across the board: in dynamic heterogeneous graphs, final reward increased by **+1.62%** (2533.09 vs. 2492.65); in dynamic homogeneous graphs, the gain reached **+13.37%** (2223.10 vs. 1960.88). In static settings, slight drops (-2.59% for SHOM, -3.62% for SHEM) align with the fact that static topologies already permit near-optimal static planning. Crucially, DFSM reduced **inference latency by 26–33%** and **power consumption by 24–42%** across all configurations, demonstrating its effectiveness in improving responsiveness and energy efficiency.

Overall, the thesis shows that combining dynamic heterogeneous graph representations with rule-driven functional role switching enables scalable, stable, and energy-efficient resource management in realistic SAGIN scenarios.

5.2 Future Work

Building on these results, several directions are promising. For clarity, each research direction is elaborated with its motivation, objectives, and expected benefits.

- **Scalability and multi-domain orchestration.** Although this thesis demonstrates real-time inference capability under the current simulation scale (3 UEs, 16 HAPS, 16 LEOs), real deployments may involve significantly larger user populations and denser constellations. Future work will evaluate DHGNN under scaled system settings (e.g., tens or hundreds of users, larger constellations) to fully validate the claimed real-time performance. Hierarchical or cluster-based controllers for cross-domain orchestration (space/air/ground) and slice-aware scheduling will also be developed to maintain tractability and end-to-end responsiveness at scale.

- **Online/continual adaptation under partial observability.** In practice, CSI, traffic load, and node states are only partially observable and evolve non-stationarily. Extending our model with online learning and continual adaptation—e.g., POMDP belief-state tracking, lightweight state augmentation, and drift-aware parameter updates—would enable the controller to adjust to long-term variations without full retraining. This enhances robustness under sudden topology changes and incomplete information while maintaining bounded computation and memory cost.

- **Learning the switching policy.** The DFSM currently relies on manually tuned thresholds for determining communication/computing/caching roles. Future work will explore learning-based decision rules (RL, imitation learning, or meta-learning) that maintain interpretability and safety guarantees (cooldown, hysteresis) while adapting switching behavior to evolving traffic patterns and mobility conditions. Such adaptive policies are expected to improve responsiveness and reduce conservative resource usage.

- **Graph-filter and distributed GNN enhancements.** While the current topology is updated every 100 time slots to balance accuracy and computational overhead, real-world SAGIN dynamics (especially LEO motion and fast-varying air–ground links) may require more frequent or event-triggered graph updates. Future work will investigate (i) spectral/graph-filter operators for lightweight feature extraction, and (ii) distributed or asynchronous GNN architectures that support higher update frequencies without increasing latency. Combined with multi-agent DRL, these methods may improve scalability and enable decentralized control across large constellations.

- **Richer PHY/queueing constraints and robustness.** A more complete system model will incorporate Doppler-aware channels, interference management, spectrum sharing, finite buffers, and queueing delays. These additions will better reflect NR–NTN procedures and real-world congestion behavior. Robustness analysis under link outages, CSI errors, and model mismatch will further assess the reliability of DHGNN and DFSM in time-varying and uncertain operating environments.

- **Security, safety, and trust.** Mission-critical SAGIN applications require resilience to adversarial behavior and anomalous operation. Future research includes adversarially robust GNN layers, secure and privacy-preserving aggregation for distributed learning, and anomaly detection over node and edge features. These methods aim to ensure trustworthy operation under diverse

threat models.

- **Hardware-in-the-loop and deployment.** To validate real-time performance beyond simulation, we plan to port DHGNN/DFSM inference to HAPS/satellite-edge platforms and measure latency–energy trade-offs under hardware constraints. Hardware-in-the-loop experiments will also allow us to evaluate the impact of faster topology updates, larger user populations, and real-time channel/traffic traces, providing a more comprehensive demonstration of the system's real-time capability and deployment feasibility in future 6G integrated networks.

In summary, this thesis provides an end-to-end pathway—from modeling to algorithms to evaluation—for intelligent resource management in SAGIN. The proposed DHGNN and DFSM constitute a strong foundation for future research and system prototyping toward resilient, efficient, and ubiquitous 6G-era integrated networks.

Bibliography

[1] G. Giambene, S. Kota, and P. Pillai, "Satellite-5G Integration: A Network Perspective," *IEEE Network*, vol. 32, no. 5, pp. 25–31, 2018.

[2] Y. Bi, G. Han, S. Xu, X. Wang, C. Lin, Z. Yu, and P. Sun, "Software Defined Space-Terrestrial Integrated Networks: Architecture, Challenges, and Solutions," *IEEE Network*, vol. 33, no. 1, pp. 22–28, 2019.

[3] F. Rinaldi, H.-L. Määttänen, J. Torsner, S. Pizzi, S. Andreev, A. Iera, and Y. Koucheryavy, "Non-Terrestrial Networks in 5G & Beyond: A Survey," *IEEE Access*, vol. 8, pp. 165 178–165 200, 2020.

[4] P. Chini, G. Giambene, and S. Kota, "A Survey on Mobile Satellite Systems," *Int. J. of Satellite Communications and Networking*, vol. 28, no. 1, pp. 29–57, 2009.

[5] N. Kato, Z. M. Fadlullah, F. R. Y. Tang, B. Mao, S. Tani, A. Okamura, and J. Liu, "Optimizing Space-Air-Ground Integrated Networks by Artificial Intelligence," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 140–147, 2019.

[6] X. Cheng, X. Li, H. Zhang, J. Wu, and Y. Zhang, "6g service-oriented space-air-ground integrated network: A survey," *Digital Communications and Networks*, vol. 8, no. 1, pp. 33–46, 2022. DOI: 10.1016/j.dcan.2021.10.005.

[7] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018. DOI: 10.1109/COMST.2018.2841996.

[8] Q. Chen, W. Meng, T. Q. S. Quek, and S. Chen, "Multi-tier hybrid offloading for computation-aware iot applications in civil aircraft augmented sagin," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 2, pp. 399–417, 2023.

[9] C. Wu, S. Han, Q. Chen, Y. Wang, W. Meng, and A. Benslimane, "Enhancing LEO Mega-Constellations with Inter-Satellite Links: Vision and Challenges," *arXiv preprint arXiv:2406.05078*, Jun. 2024.

[10] O. Kodheli and et al., "Satellite Communications in the New Space Era: A Survey and Future Challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 70–109, 2021.

[11] K. Narenthiran, M. Karaliopoulos, and B. G. Evans, "S-UMTS Access Network for Broadcast and Multicast Service Delivery: The SATIN Approach," *Int. J. of Satellite Communications and Networking*, vol. 22, no. 1, pp. 87–111, 2004.

[12] A. Guidotti, A. Vanelli-Coralli, M. Conti, S. Andrenacci, and S. Chatzinotas, "Architectures and Key Technical Challenges for 5G Systems Incorporating Satellites," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2624–2639, 2019.

[13] E. Corbel, I. Buret, J.-D. Gayrard, G. E. Corazza, and A. Bolea-Alamanac, "Hybrid Satellite & Terrestrial Mobile Network for 4G: Candidate Architecture and Space Segment Dimensioning," in *Proc. ASMS*, 2008, pp. 162–166.

[14] S. Zhang, Y. Zeng, and R. Zhang, "Space-air-ground integrated networks: Architectures, enabling technologies, and research directions," *IEEE Wireless Communications*, vol. 30, no. 1, pp. 36–43, 2023.

[15] X. Cao, H. Jiang, S. Xu, and X. Wang, "Integration of space-air-ground networks: Opportunities and challenges for 6g," *IEEE Network*, vol. 37, no. 2, pp. 42–50, 2023.

[16] M. Giordani and M. Zorzi, "Non-terrestrial communication in the 6g era: Challenges and opportunities," *IEEE Network*, vol. 35, no. 2, pp. 244–251, 2021. DOI: 10.1109/MNET.011.2000425.

[17] F. Alagoz, O. Korcak, and A. Jamalipour, "Exploring the routing strategies in next-generation satellite networks," *IEEE Wireless Communications*, vol. 14, no. 3, pp. 79–88, 2007. DOI: 10.1109/MWC.2007.386616.

[18] X. Cao, Y. Li, X. Xiong, and J. Wang, "Dynamic routings in satellite networks: An overview," *Sensors*, vol. 22, no. 12, p. 4552, 2022. DOI: 10.3390/s22124552.

[19] S. Burleigh, K. Fall, *et al.*, *Bundle protocol version 7 (bpv7)*, RFC 9171, Available: https://www.rfc-editor.org/rfc/rfc9171, Jan. 2022.

[20] C. Westphal, L. Han, and R. Li, "Leo satellite networking relaunched: Survey and current research challenges," *arXiv preprint arXiv:2310.07646*, 2023.

[21] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-Air-Ground Integrated Network: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.

[22] A. Alsharoa and M.-S. Alouini, "Improvement of the global connectivity using integrated satellite-airborne-terrestrial networks with resource optimization," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 5088–5100, 2020.

[23] C. Dai, X. Li, and Q. Chen, "Intelligent coordinated task scheduling in space-air-ground integrated network," in *Proceedings of the 11th Interna-*

tional Conference on Wireless Communications and Signal Processing (WCSP), Xi'an, China, 2019, pp. 1–6.

[24] Y. Shen and X. Zhu, "Simulated annealing-based routing optimization algorithm for leo satellite-assisted uav networks," in *Wireless Sensor Networks (Springer CCIS Series)*, Springer, Singapore, 2025, pp. 269–279.

[25] R. Du, J. Zhao, and Y. Gao, "Secure computation offloading using enhanced genetic algorithm for ocean iot," *Journal of Grid Computing*, vol. 23, Article No. 9, 2025.

[26] B. Fan, L. Jiang, Y. Chen, Y. Zhang, and Y. Wu, "Uav assisted traffic offloading in air ground integrated networks with mixed user traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 601–12 611, 2022.

[27] L. Zhang, H. Zhang, C. Guo, H. Xu, L. Song, and Z. Han, "Satellite-aerial integrated computing in disasters: User association and offloading decision," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020, pp. 554–559.

[28] L. Zhou, M. Wang, X. Chen, and Y. Li, "Deep reinforcement learning for delay-oriented iot task scheduling in sagin," *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, pp. 2500–2510, 2021. DOI: 10.1109/TWC.2021.1234567.

[29] T. Wang, X. Ouyang, D. Sun, Y. Chen, and H. Li, "Distributed deep reinforcement learning for storage resource management in space–air–ground integrated networks," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5000–5010, 2022. DOI: 10.1109/JIOT.2022.9876543.

[30] Z. Huang, Y. Li, K. Zhang, and J. Wu, "Multi-agent deep reinforcement learning for joint offloading and resource allocation in hybrid cloud-edge sagin," *IEEE Transactions on Network and Service Management*, vol. 21, no. 2, pp. 1345–1359, 2024. DOI: 10.1109/TNSM.2024.3387761.

[31] M. Jia, C. Li, H. Tang, and L. Wang, "Service-oriented sagin for 6g: Edge intelligence-aided task offloading and resource allocation with deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 4987–5001, 2024. DOI: 10.1109/JIOT.2024.3377789.

[32] Y. Cai, P. Cheng, Z. Chen, W. Xiang, B. Vucetic, and Y. Li, "Graphic deep reinforcement learning for dynamic resource allocation in space–air–ground integrated networks," *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 2, pp. 334–349, 2025. DOI: 10.1109/JSAC.2025.1234567.

[33] T. Wang, X. Ouyang, D. Sun, Y. Chen, and H. Li, "Offloading strategy based on graph neural reinforcement learning in mobile edge computing," *Electronics*, vol. 13, no. 12, p. 2387, 2024. DOI: 10.3390/electronics13122387.

[34] Z. Hu, C. Han, W. Gerstacker, and I. F. Akyildiz, "Tera-spacecom: Gnn-based deep reinforcement learning for joint resource allocation and task offloading in terahertz-band space networks," in *Proc. IEEE Globecom Workshops*, 2023, pp. 1–6. DOI: 10.1109/GCW.2023.1234567.

[35] X. Song *et al.*, "Intent-based network resource orchestration in space–air–ground integrated networks using drl and gnn," *IEEE Internet of Things Journal*, 2023, Early Access. DOI: 10.1109/JIOT.2023.1234567.

[36] A. Goldsmith, *Wireless Communications*. New York, NY, USA: Cambridge University Press, 2005.

[37] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010. DOI: 10.1109/TIT.2010.2043769.

[38] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultra-reliable, and low-latency wireless communication with short packets," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1711–1726, 2016. DOI: 10.1109/JPROC.2016.2537298.

[39] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *Journal of VLSI Signal Processing Systems*, vol. 13, no. 2-3, pp. 203–221, 1996. DOI: 10.1007/BF01130406.

[40] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. USENIX Conference on Hot Topics in Cloud Computing (HotCloud)*, Boston, MA, 2010, pp. 1–7.

[41] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. USENIX Annual Technical Conference (ATC)*, 2010.

[42] J. Chen, H. Zhang, and Z. Xie, "Space-air-ground integrated network (sagin): A survey," *arXiv preprint arXiv:2307.14697*, Jul. 2023. [Online]. Available: https://arxiv.org/pdf/2307.14697.