Intelligent Computing
A SCIENCE PARTNER JOURNAL

REVIEW ARTICLE

Space Computing: Architectures, Challenges, and Future Directions

Erzhong Xue[1†], Zhuoran Zhang[1†], Junxiao Xue[1], Haitao Wang[1], Ivan E. Carvajal-Roca[2], Zhiwen He[3], Hui Zhang[1], Hua Wang[1], Zhiguo Wan[1], and Chao Li[1*]

[1]Center for Space Computing, Zhejiang Lab, Hangzhou, China. [2]Tianjin University, Tianjin, China. [3]South China Normal University, Guangzhou, China.

*Address correspondence to: lichao@zhejianglab.org
†These authors contributed equally to this work.

Citation: Xue E, Zhang Z, Xue J, Wang H, Carvajal-Roca IE, He Z, Zhang H, Wang H, Wan Z, Li C. Space Computing: Architectures, Challenges, and Future Directions. *Intell. Comput.* 2025;4:Article 0134. https://doi.org/10.34133/icomputing.0134

Submitted 23 April 2025
Revised 23 August 2025
Accepted 27 August 2025
Published 14 November 2025

In recent years, the rapid advancement of space technologies has given rise to sophisticated space computing systems, which have become crucial for various applications such as Earth observation, communication, and scientific research. This survey paper provides a comprehensive overview of the history, current state, and future trends in space computing systems. We discuss the evolution of these systems, highlighting important milestones from early satellite communication networks to modern space–air–ground integrated networks (SAGIN). Despite their advancements, space computing systems face numerous challenges, including resource constraints and unstable network links. To address these challenges, we explore key enabling technologies that are critical for the future of space computing. These include virtualization and resource pooling, which enhance resource utilization; distributed storage, which provides resilience and efficiency; advanced scheduling and orchestration techniques, which optimize task allocation across heterogeneous resources; collaborative AI training and inference, which leverage distributed satellite networks for real-time data processing; and robust security and privacy measures, which are vital for safeguarding sensitive information in space operations. By examining existing systems and identifying open research challenges, this paper aims to provide a road map for future research and development in the field of space computing systems. Our findings highlight the potential of emerging technologies to transform space operations and underscore the importance of continued innovation in overcoming current limitations.

Introduction

As mega-constellations of smart satellites, operated by entities such as SpaceX, OneWeb, and Telesat, proliferate, they unleash a flood of sensor data from space and promise an interconnected network stretching from the Earth's surface to space. Against this backdrop, space computing is emerging as a potential technological revolution. Space computing systems integrate advanced computing technologies and architectures tailored for space environments. These systems demand a fusion of space-hardened hardware–processors, storage, memory, routers, and software-defined distributed computing systems to meet various and evolving mission needs. They link a network of satellites, space stations, and other space assets to support applications as diverse as telecommunications, Earth observation, navigation, and scientific exploration. The unique value of space computing lies in its ability to provide global coverage, enable real-time data acquisition, and facilitate observations and communications that are unattainable through terrestrial systems. For example, space systems are indispensable in weather forecasting, disaster management, environmental monitoring, and global communication networks. By continuously collecting and transmitting data from space, these systems not only deepen our understanding of the Earth and the universe but also bolster essential services that shape daily life.

History of space computing systems and challenges

The evolution of space computing systems is closely related to the history of space exploration. Since the launch of Sputnik 1 in 1957, space missions have increasingly relied on onboard computers for navigation and communication. The complexity of onboard computing grew with missions like the Apollo program in the 1960s, which relied on advanced computers for navigation and lunar landing operations.

The 1970s and 1980s saw the deployment of space stations such as Skylab and the Mir space station, which utilized more sophisticated computing systems to manage life support, scientific experiments, and data transmission. The launch of the Hubble Space Telescope in 1990 further expanded the role of space computing, with onboard systems capable of processing vast amounts of astronomical data. The International Space Station (ISS), operational since 1998, represents an important milestone in space computing, featuring a network of computers

that manage a range of functions, from scientific research to communication with Earth.

In recent years, the focus has shifted to space–air–ground integrated networks (SAGIN) and satellite constellations like SpaceX's Starlink, which aim to provide global high-speed internet coverage. The advent of CubeSats and small satellites has also democratized space access, fueling interest in commercial applications of space computing, such as Earth observation for agriculture, disaster response, and environmental monitoring.

Despite these advancements, space computing systems face persistent challenges, notably resource limitations. Satellites operate in an environment with limited power, computational capacity, and storage. Harsh space conditions necessitate specialized hardware, complicating system design and operation. Additionally, unstable network links due to satellite motion and vast distances result in intermittent communication, causing data loss, increased latency, and difficulties in maintaining continuous operations. For example, low Earth orbit (LEO) satellites' frequent handovers over different ground stations complicate network management and data transmission.

Key technologies in space computing systems

To address these challenges, several key technologies are being developed and implemented in space computing systems.

1. Virtualization and pooling: Virtualization allows multiple virtual machines (VMs) to run on a single physical machine, optimizing the use of available resources. In space computing, virtualization can facilitate the dynamic allocation of computational resources, enabling more efficient processing and management of tasks. Pooling of resources, such as processing power and memory, across a network of satellites can enhance overall system performance and reliability.
2. Distributed storage: The vast amount of data generated by space sensors necessitates efficient storage solutions. Distributed storage systems, which spread data across multiple nodes, offer scalability, reliability, and faster access. These systems can store large datasets, such as satellite imagery and telemetry data, while ensuring data redundancy and fault tolerance.
3. Scheduling and orchestration: Effective scheduling and orchestration of tasks are crucial for managing the limited resources available in space. Advanced algorithms and frameworks are used to prioritize tasks, allocate resources, and manage workloads across multiple satellites and ground stations. Scheduling and orchestration are especially important for missions that involve real-time data processing and time-sensitive operations.
4. Collaborative artificial intelligence (AI) training and inference: The integration of AI in space computing systems can enhance data analysis, decision-making, and automation. Collaborative AI training, which leverages data from multiple satellites and ground-based sources, can enhance the accuracy and robustness of AI models. AI inference, conducted onboard satellites, enables real-time analysis and decision-making, reducing the need for data transmission to Earth.
5. Security and privacy: Given the critical nature of space systems, ensuring security and privacy is paramount. Space computing systems are vulnerable to cyberattacks, data breaches, and unauthorized access. Implementing robust encryption, authentication, and access control measures is essential to protect sensitive data and maintain the integrity of these systems. Additionally, secure communication protocols and data privacy frameworks are necessary to safeguard information transmitted between satellites and ground stations.

Contribution and scope of this review

This review aims to provide a comprehensive analysis of the architectures, challenges, and key technologies in space computing systems. It distinguishes itself from previous surveys by emphasizing the latest advancements in AI integration, distributed systems, and security measures. By examining the current state of the art and exploring future research opportunities, this review offers new insights into the evolving landscape of space computing. The discussion includes an exploration of cooperative computing frameworks, blockchain-enabled collaborative systems, and the potential for large-scale AI model training and inference in space. Through this, we aim to highlight the critical role of space computing in advancing scientific research, enhancing global communication, and addressing the challenges of our rapidly changing world. For convenience, some main abbreviations used in this paper are listed in Table 1. A comparison table summarizing the similarities and differences between this work and several highly cited survey manuscripts from the past few years is presented in Table 2. The structure of the survey is presented in Fig. 1.

Space Computing Architectures and Related Technologies

The satellite Internet of Things (IoT) is a key application area of both SAGIN and satellite terrestrial network (STN), as it utilizes satellite networks to provide global connectivity for IoT devices. Satellite IoT takes advantage of the extensive coverage of satellites to offer communication services for IoT devices in remote areas, over oceans, and in the air, where terrestrial network access is difficult. For instance, in smart agriculture, satellite IoT can monitor soil moisture and crop growth in real time; in maritime transportation, it can track the location and status of ships and cargo.

Space–air–ground integrated networks

SAGIN is a comprehensive communication architecture that integrates ground networks, aerial devices (such as drones and high-altitude balloons), and space networks (including satellite systems) [1]. Its goal is to provide seamless global communication services to users in space, the air, at sea, and on the ground. By integrating various network resources, SAGIN offers full-time, all-domain, and all-space communication and connectivity services, meeting the future needs for high bandwidth, low latency, and wide coverage (see Fig. 2).

Ground networks

Ground networks facilitate terrestrial data communication through systems like cellular networks, optical fiber, local area networks (LANs), and mobile ad hoc networks. As the cornerstone of ground networks, cellular networks provide wireless connectivity to a vast user base, especially in densely populated areas. Advancements in technologies like

Table 1. List of major abbreviations

Abbreviation	Full form	Abbreviation	Full form
ACO	Ant colony optimization	AE	Authenticated encryption
AES	Advanced encryption standard	AI	Artificial intelligence
B5G	Beyond 5G	CDN	Content delivery network
CNN	Convolutional neural network	DBMS	Database management system
DFS	Distributed file system	DRL	Deep reinforcement learning
FL	Federated learning	GEO	Geostationary Earth orbit
GPU	Graphics processing unit	HAP	High-altitude platform
ICN	Information-centric networking	IoT	Internet of Things
LAN	Local area network	LAP	Low-altitude platform
LEO	Low Earth orbit	LSTM	Long short-term memory
MDS	Maximum distance separable (codes)	MEO	Medium Earth orbit
NFV	Network function virtualization	NTN	Nonterrestrial network
PCI	Peripheral component interconnect	QKD	Quantum key distribution
QoE	Quality of experience	RS	Reed–Solomon (codes)
SAGIN	Space–air–ground integrated network	SAGSI	Space–air–ground–sea integrated network
SDN	Software defined networking	SR-IOV	Single-root I/O virtualization
STECN	Satellite-terrestrial integrated edge computing network	STN	Satellite-terrestrial network
TN	Terrestrial network	UAV	Unmanned aerial vehicle
VM	Virtual machine		

multiple-input multiple-output (MIMO) and network slicing have ushered in the 5G era, characterized by high-speed, low-latency connectivity. 5G networks are now commercially available. Research now focuses on the next generation of mobile networks, beyond 5G (B5G) and 6G, which is expected to revolutionize telecommunications.

Aerial networks

Aerial networks provide mid-layer wireless communication services within the SAGIN framework. They are realized through 2 complementary altitude classes: low-altitude platforms (LAPs), typically rotor-wing unmanned aerial vehicles (UAVs) operating between 100 m and 10 km, and high-altitude platforms (HAPs), such as solar UAVs or stratospheric balloons cruising between 17 and 30 km. LAPs are deployed on demand to cover local hotspots or disaster areas with ultra-low latency, while HAPs act as quasi-stationary "pseudo-satellites", offering wide-area footprints and latency on par with fiber. Together, these platforms extend the reach of wireless services more economically than ground infrastructure alone, and by integrating lightweight multi-access edge computing (MEC) payloads, they can cache popular content or run AI inference before traffic ever reaches a satellite.

Space networks

Space networks are composed of satellite systems, which include both the satellites and the elements of their corresponding terrestrial infrastructure, such as ground stations and network operation centers. Satellites are categorized by their orbits into geostationary Earth orbit (GEO), medium Earth orbit (MEO), and LEO. GEO satellites match the Earth's rotation rate, appearing stationary over a fixed point and offering extensive coverage, with a trio of such satellites nearly able to blanket the globe, excluding the polar regions. MEO satellites, like Inmarsat-P and Odyssey, are widely used for navigation and communication services. LEO satellites, which have lower latencies and higher data transmission speeds compared to their GEO and MEO counterparts, are becoming increasingly prevalent. Advances in satellite manufacturing and launch technologies have enabled the deployment of numerous LEO satellites to form constellations. Companies such as SpaceX, OneWeb, and Amazon are planning to launch extensive LEO constellations comprising thousands of satellites to deliver high-throughput broadband services worldwide.

Satellite-terrestrial integrated edge computing networks

STNs, or satellite-terrestrial integrated systems, are an important component of SAGIN, focusing on the integration of satellite and terrestrial systems. They provide wide-area coverage and highly reliable services through communication links between satellites and ground stations. STNs leverage the extensive coverage of satellites to complement the shortcomings of terrestrial networks in remote areas or during natural disasters. For example, STNs can serve as backups for terrestrial networks, providing emergency communication services when earthquakes or floods damage ground communication infrastructure.

By deploying MEC in STNs, some basic applications can be realized in the network architecture, such as content caching, computation offloading, and network services. These applications expand the function of STNs and improve their capability. While theoretically, satellites can be viewed as nodes with

Table 2. Comparison between our survey and recent high-impact SAGIN surveys

Reference	Scope	Main technical focus	Main contribution	Distinctive contributions of our survey
[138]	6G SAGIN	Communication and computing convergence, PHY/MAC/network layers	Comprehensive 6G SAGIN survey emphasizing integrated communication and computing	We further dissect computing subsystems: distributed storage, GPU virtualization, AI training/inference, and security tailored to space constraints
[1]	SAGIN	Network design, resource allocation, performance optimization	First holistic survey on SAGIN	We add (a) distributed storage and erasure coding, (b) GPU/virtualization and scheduling, (c) AI distributed training/inference, (d) system-level security and privacy
[139]	SAGSI	Security threats, attack models, countermeasures	First security survey covering full SAGSI	We focus on computing-system security (encryption, authentication, key management) and AI-era threats
[140]	SAGSI	Resource optimization (throughput, delay, energy, off-loading)	Taxonomy of resource-optimization techniques	We concentrate on computing-resource virtualization, scheduling, and AI collaboration rather than radio-resource optimization
[141]	6G SAGIN	Virtualized/softwarized 6G architecture	Proposes flat, low-latency 6G SAGIN architecture	We do not repropose architecture; instead, details how to run efficient, secure, AI-ready computing services atop 6G SAGIN
[142]	6G SAGIN	Service-oriented management, cloud-edge synergy	Service-oriented SAGIN management framework	We treat computing services (AI training/inference, distributed storage) as first-class 6G services and details their enablers
[143]	6G SAGIN	Key 6G enablers (UAV, satellite)	Vision of 6G-empowered SAGIN	We systematize computing enablers: virtualization, scheduling, AI, storage, security
[144]	6G NTN	NTN/TN integration, architectures, use cases	Evolution roadmap of NTN in 5G/6G	We highlight how computing capabilities evolve with NTN and provides AI/virtualization solutions
[145]	SAGIN	Edge/quantum/distributed computing	Survey of advanced computing for SAGIN	We further detail: distributed storage, AI collaborative learning, model compression, secure computing
[2]	STECN	Multi-layer edge computing architecture	First STECN architecture	We position STECN inside SAGIN and augment with AI training/inference, model compression, security mechanisms

processing capabilities, existing research on MEC in STNs predominantly treats satellite networks as relay networks, overlooking their potential for direct task processing. Design issues for multi-layered edge computing architectures and coordination of heterogeneous edge computing resources are also underexplored in traditional STNs. To address these limitations and enhance service and application support in future networks, the satellite-terrestrial integrated edge computing network (STECN) architecture has been proposed [2] (see Fig. 3).

The core control functionalities of STECN are realized by the STECN platform, which employs various modules to orchestrate the entire network efficiently. An STECN typically comprises a satellite network, terrestrial network, edge computing clusters, and user devices.

- The satellite network often involves LEO satellites equipped with MEC platforms to handle computational tasks from user devices.
- Terrestrial platforms include cellular networks, backbone networks, data centers, and terrestrial MEC platforms that might host cloud-based platforms (e.g., cloudlets). These terrestrial MEC platforms, with more computational resources than satellites and clusters, can process tasks from user devices, communicate via cellular networks, and relay tasks to data centers through backbone networks and the internet.
- Edge computing clusters, i.e., maritime, aerial, and terrestrial (vehicle) clusters, are collections of devices with ample computational resources. Equipped with MEC platforms, they process tasks from user devices and enhance efficiency through task collaboration.
- User devices include smartphones, augmented reality and virtual reality devices, and intelligent vehicles. These devices have poor computational capability, so they generate computation tasks and require STECN to handle them.

STECNs leverage software-defined networking (SDN) and network function virtualization (NFV) technologies to virtualize compute, storage, and networking resources. Resource virtualization facilitates basic resource invocation and interfaces with upper-level control planes, enabling unified management and improving resource utilization while simplifying network scalability and maintenance. Signal separation techniques and mobility management are also crucial technologies within the STECN framework.

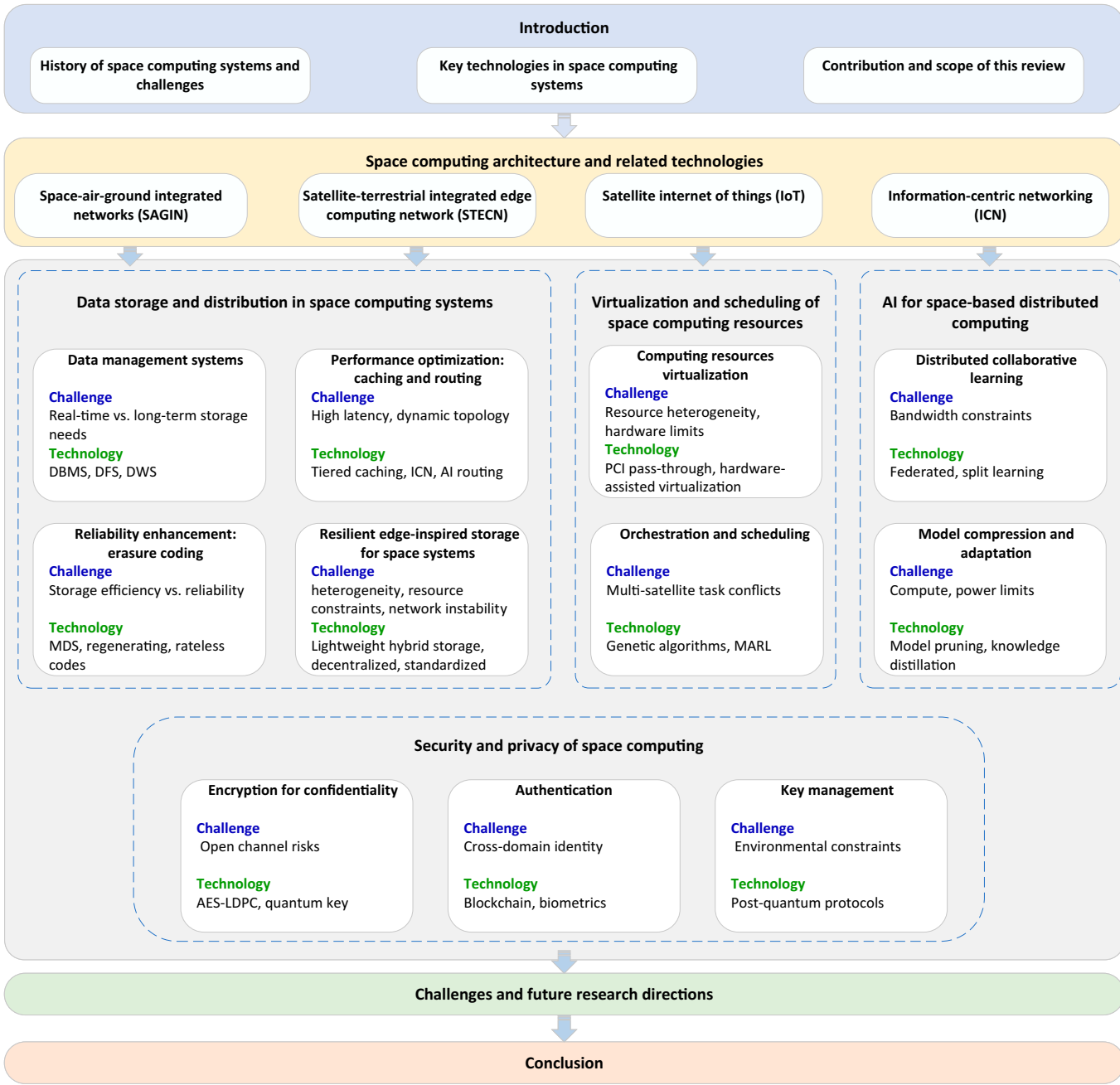

Fig. 1. The structure of the survey.

Satellite IoT

The satellite IoT represents a sophisticated integration of space assets with ground-based IoT devices, creating a global network capable of connecting remote, mobile, and even previously unconnected objects. The architecture of satellite IoT is designed to overcome the limitations of terrestrial networks and provides a robust framework for data transmission, processing, and management across a wide range of applications.

The architecture of the satellite IoT is multi-layered, consisting of the following key components (see Fig. 4):

- Sensors and IoT devices: These are the ground-level components that collect data from various environments and systems. They can be deployed in remote sensing stations, agricultural fields, and wildlife habitats, and even in mobile assets like vehicles and ships.
- Ground stations: These are the Earth-based facilities responsible for communicating with IoT devices and the satellite network. Ground stations play a crucial role in data transmission, acting as the gateway between the IoT devices and the satellite constellations.
- Satellite constellations: Comprising a series of LEO, MEO, and GEO satellites, these constellations provide the backbone of the satellite IoT network, offering global coverage and data relay services.
- Satellite IoT gateways: These are specialized devices or systems located on satellites that receive data from

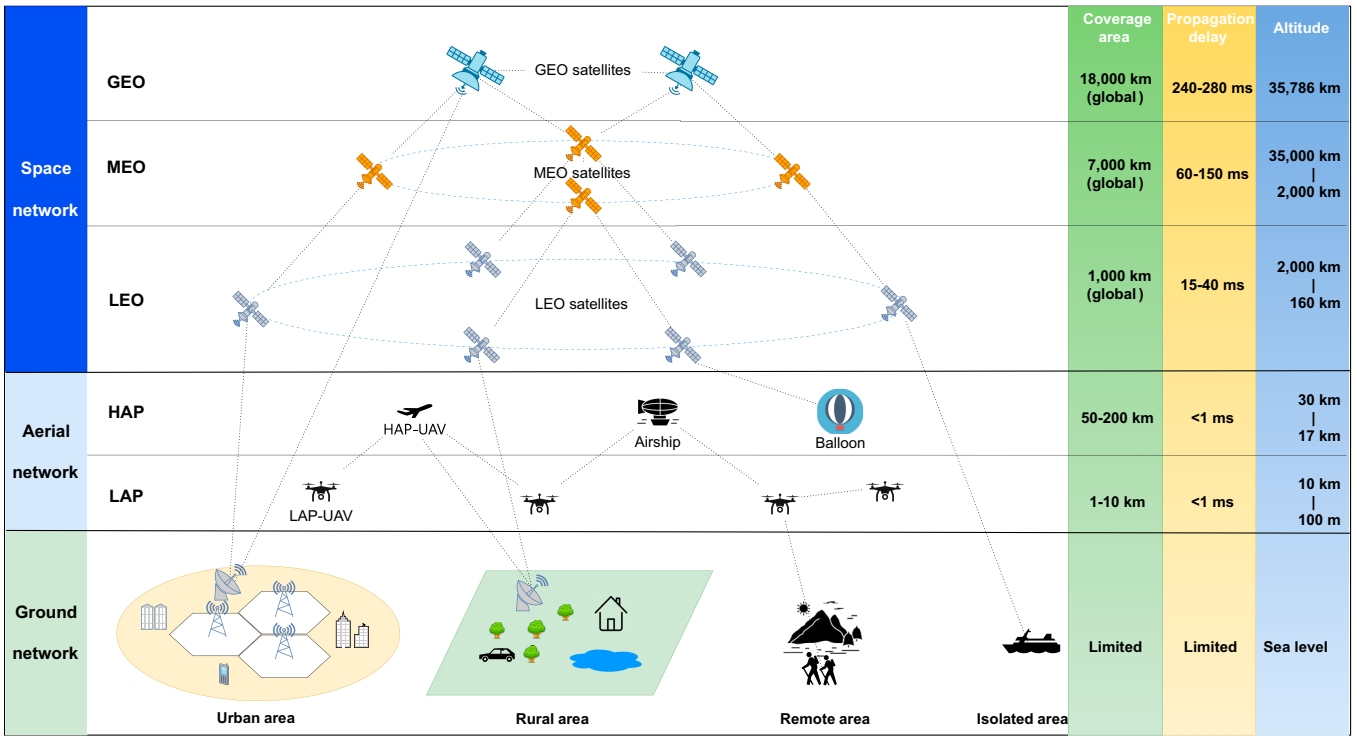

Fig. 2. Architecture of SAGINs.

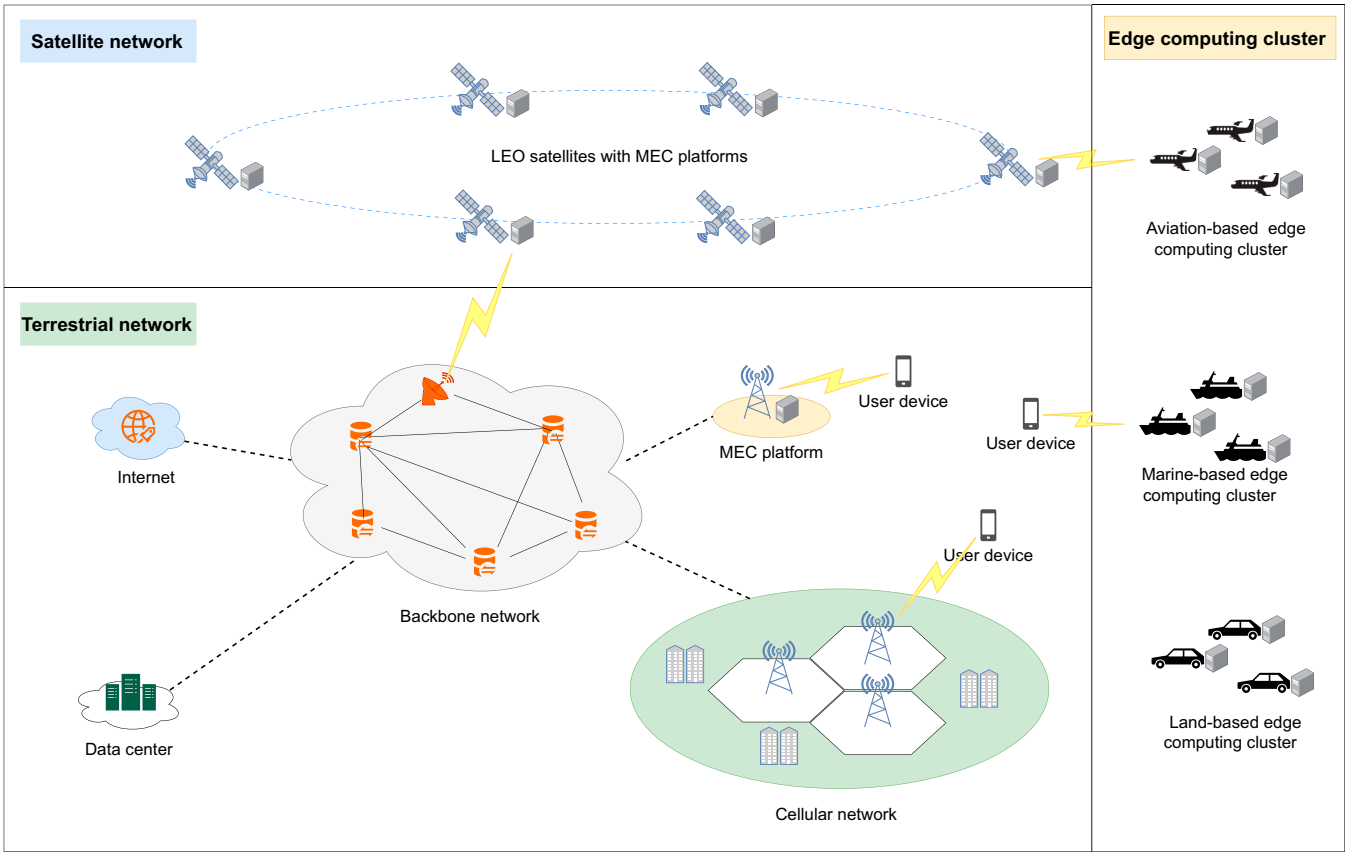

Fig. 3. Architecture of STECN.

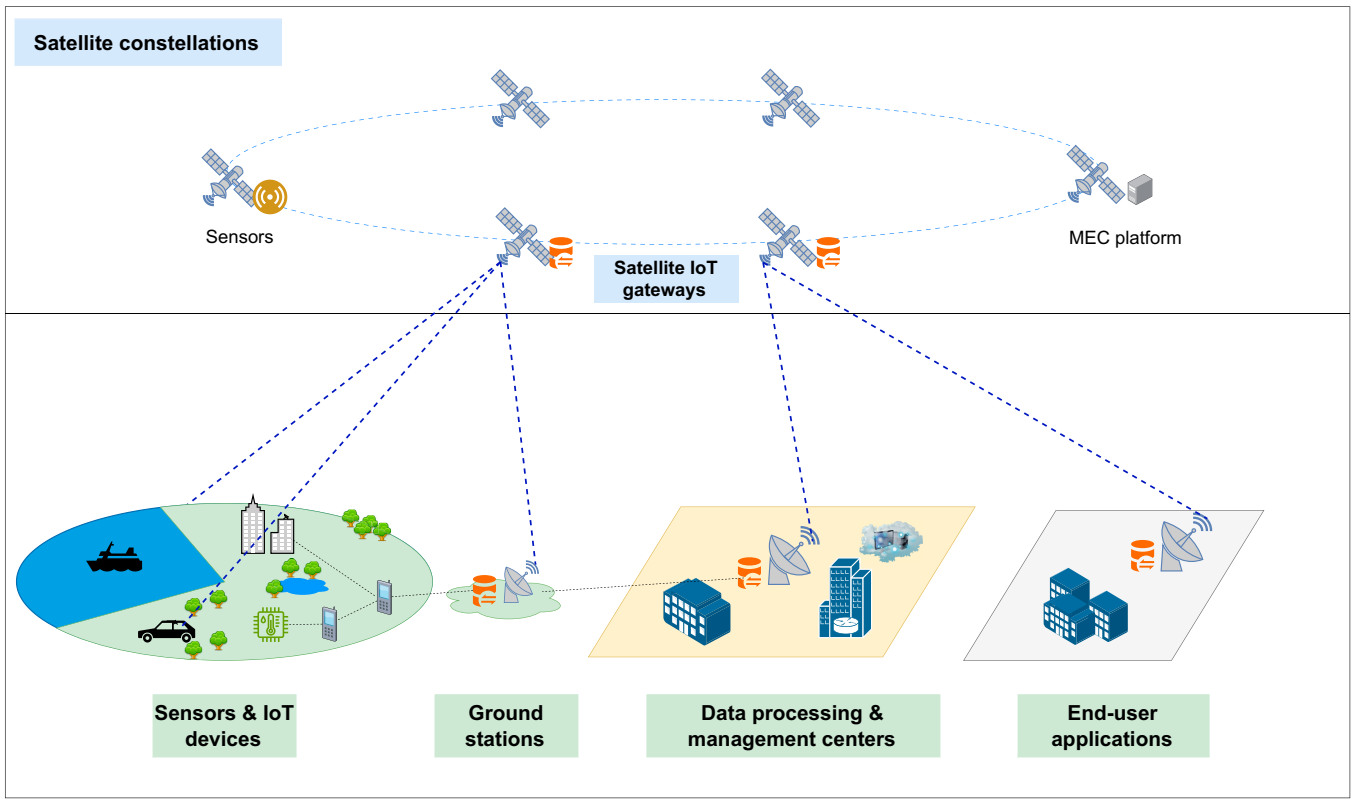

Fig. 4. Architecture of satellite IoT.

ground stations and relay it to other parts of the network or directly to end-user systems.

- Data processing and management centers: Once data are collected and transmitted to the processing centers, they can be analyzed and managed using cloud computing platforms, machine learning (ML) algorithms, and database management systems (DBMSs).
- End-user applications: These are the systems or platforms that utilize the processed data for applications ranging from environmental monitoring to asset tracking and smart agriculture.

One critical aspects of satellite IoT architecture is the integration and interoperability of different components and systems. This ensures seamless data flow and allows for the customization of services to meet specific application needs.

Satellite IoT architecture is designed to be scalable and flexible to allow for the addition of more IoT devices, ground stations, or satellites as the network expands or as demand increases.

Given the distributed nature of the satellite IoT network, robust security measures are implemented at all levels to protect against unauthorized access, data breaches, and other cyber threats. Data privacy is also a key consideration, with encryption and secure data handling practices in place.

In summary, the architecture of the satellite IoT is a sophisticated blend of space and terrestrial technologies, designed to provide global connectivity for IoT devices. Its components work in concert to collect, transmit, process, and utilize data, offering a powerful platform for a wide range of applications that can benefit from the unique capabilities of satellite-enabled IoT solutions.

Information-centric networking

In recent years, by focusing on content and integrating existing content delivery networks (CDNs), information-centric networking (ICN) has gradually been established. Unlike to traditional connection-centric networks, ICN adopts a publish and subscribe model, enabling more effective content-aware routing strategies. Two important features of ICN are in-network caching and routing by name.

ICN focuses on content processing and distribution rather than host-to-host communication, shifting content transmission from a sender-driven end-to-end communication model to a receiver-driven massive content acquisition model, reducing information transmission latency.

However, there are 2 issues: The mobility of satellite nodes requires adjustments in the return path in the transmission path, leading to errors in the pending interest table (PIT) and the forward information base (FIB), making content packets unable to find the requesting nodes. Such path changes can cause shifts in the location of content copy caching, reducing caching benefits. In addition, traditional caching strategies in the ICN network have low caching benefits, while new caching strategies have high overhead and are not suitable for satellite node transmission.

Data Storage and Distribution in Space Computing Systems

In terrestrial distributed storage systems, optimizing data management and storage strategies is crucial to ensure performance, reliability, and scalability. These systems face several challenges, including handling massive amounts of data, achieving high

throughput, and ensuring high availability. To address these challenges, advanced file systems and DBMSs are commonly employed, along with redundancy and replication mechanisms to enhance fault tolerance and parallel processing efficiency [3,4]. However, when these storage systems are applied in space-based distributed computing environments, new challenges arise, including intermittent connectivity, limited bandwidth, and higher latency [5,6]. These unique constraints necessitate a reevaluation and adjustment of existing storage strategies to ensure data integrity and accessibility. Consequently, space-based distributed storage systems must not only inherit the strengths of terrestrial systems but also incorporate specific technologies such as intelligent data distribution, erasure coding, and optimized routing mechanisms to reliably operate under challenging conditions.

Data management systems

In the field of data management systems, various advanced technologies and tools work together to support data storage, processing, and analysis. These include, but are not limited to, DBMSs, distributed file systems (DFSs), and data warehouse systems (DWSs). The application scenarios of these data management systems can be broadly divided into 2 categories: one is the storage of satellite data on the ground, and the other is the storage of satellite data on the satellite itself. Ground-based satellite data storage primarily focuses on the long-term management and complex analysis of vast amounts of data and thus requires capabilities to handle and store large-scale data. In contrast, space-based data storage emphasizes real-time performance, with systems needing to store, process, and transmit data in real time under limited resources and bandwidth while maintaining efficient and reliable operation under extreme environmental conditions.

Database management systems: DBMSs are a core component in the domain of data management systems. They are used to create, manage, operate, and analyze databases. The key functions of DBMSs include data definition, storage, querying, updating, security control, data backup and recovery, concurrency control, and data integrity maintenance. In space-based systems, traditional DBMSs, such as relational DBMSs (RDBMSs), ensure reliable management of large-scale structured satellite data through efficient data access and transaction processing capabilities. Nonrelational DBMSs (commonly called NoSQL systems, meaning Not only SQL systems), on the other hand, address the challenges of handling diverse and nonlinearly growing satellite data. Additionally, DFSs excel in satellite data storage applications.

Distributed file systems: In space-based storage scenarios, DFSs (such as Hadoop DFS, i.e., HDFS) provide scalable distributed storage and high-throughput data processing under extreme physical conditions, ensuring effective data management even in cases of intermittent connectivity and high latency.

Data warehouse systems: DWSs focus on centralized storage and analysis of large satellite datasets from various sources, supporting complex analysis tasks and business intelligence.

These systems each have their own characteristics and, through different technologies and architectures, address the diverse needs of space-based data management. Some key data management systems related to satellites are listed in Table 3.

While efficient data management systems ensure structured storage, the harsh conditions of space environments (e.g., intermittent connectivity) demand additional techniques to enhance data reliability and reduce transmission overhead. This motivates the use of erasure coding and caching strategies.

Reliability enhancement: Erasure coding

Traditional data storage often uses multiple file copies across different units for reliability, which is inefficient due to high storage overhead. The hardware, software, maintenance, and resource consumption costs are especially high for large-scale satellite data. Erasure coding is an effective way to reduce these overhead costs while maintaining data reliability. Another key factor in selecting erasure codes is their ability to handle failures of individual storage units, as such failures are quite common in practical operations. These failures include not only physical malfunctions of storage units but also instances where storage units become unavailable due to maintenance downtime or competing service requests. Erasure coding is a widely adopted data encoding technique in distributed storage systems that provides 3 key optimization strategies for data storage and repair in space-based environments. An overview of erasure codes for distributed storage is given in Fig. 5.

Minimization of storage

In distributed storage systems, maximum distance separable (MDS) codes have garnered significant attention due to their unique advantages. MDS codes possess the theoretically maximum minimum Hamming distance $d_{\min} = n - k + 1$, which allows them to achieve optimal distance properties given a code length n and dimension k. This distance characteristic makes MDS codes highly efficient in error correction and detection, allowing for the correction of more errors with minimal redundancy. Moreover, MDS codes are easy to implement and highly scalable, making them suitable for efficient deployment across various hardware and software systems, thereby adapting to data protection applications of varying scales. MDS codes, such as Reed–Solomon (RS) codes, have gained considerable attention due to their unique advantages in distributed storage systems. RS codes have been used in the Hadoop DFS with erasure coding (HDFS-EC) [7], Facebook's f4 BLOB storage [8], and Baidu's Atlas Cloud Storage [9].

Minimization of repair bandwidth

Designing efficient coding schemes requires consideration not only of maintaining efficiency in storage overhead but also of possessing robust capabilities for repairing failed nodes. When repairing failed nodes, data must be downloaded from auxiliary nodes. The bandwidth consumed by this data download is known as repair bandwidth. Addressing these practical needs, coding theorists have proposed 2 types of novel codes: regenerating (RG) codes and locally recoverable (LR) codes. Regenerating codes focus on minimizing repair bandwidth, and LR codes aim to minimize the number of auxiliary nodes contacted during node repairs, which is referred to as repair locality. On another front, coding theorists have also reexamined the issue of node repair in RS codes and proposed new, more efficient repair techniques. A special type of regenerating code is the minimum storage regenerating (MSR) code, which maintains the same storage efficiency as MDS codes while considerably reducing repair bandwidth. This makes MSR codes highly valuable in distributed storage systems, particularly in environments with high bandwidth costs.

Table 3. Data management systems for satellite data

Data management system	Type and examples		Key features	Application scenarios	Typical works	
					Ground segment	Satellite segment
DFS	HDFS		•Batch processing •High throughput •Highly scalable •Specialized in large-scale storage	•Reliable storage and processing •Large volumes of raw satellite data	[146–148]	[6]
DBMS	Relational DBMS	Oracle spatial	•Precise geospatial processing •Advanced data analysis •Complex spatial queries	•Geospatial data	[149]	
	Nonrelational DBMS (NoSQL)	HBase	•Real-time processing of large-scale unstructured data •Column-based storage •Fast read/write support	•Rapidly growing sensor data •Unstructured data •Fast access and real-time analysis	[146]	[6]
		Mongo DB	•Flexible document-based storage •Dynamic semi-structured data handling •Adaptive data management	•Diversity and rapid changes of satellite	[5,149]	
		Sci DB	•Optimized multi-dimensional array storage •Designed for scientific computation •Efficient data organization	•Complex scientific data •High precision calculations	[150,151]	
		Key-value	•Minimalitst design •High-performance •Low latency •Fast access	•Real-time data management •Quick retrieval		[152]
DWS	Hive		•SQL-like query support •Complex analysis handling •Report generation •Structured and semi-structured data processing	•Batch processing, querying, and analysis •Large historical satellite datasets	[5]	[6]

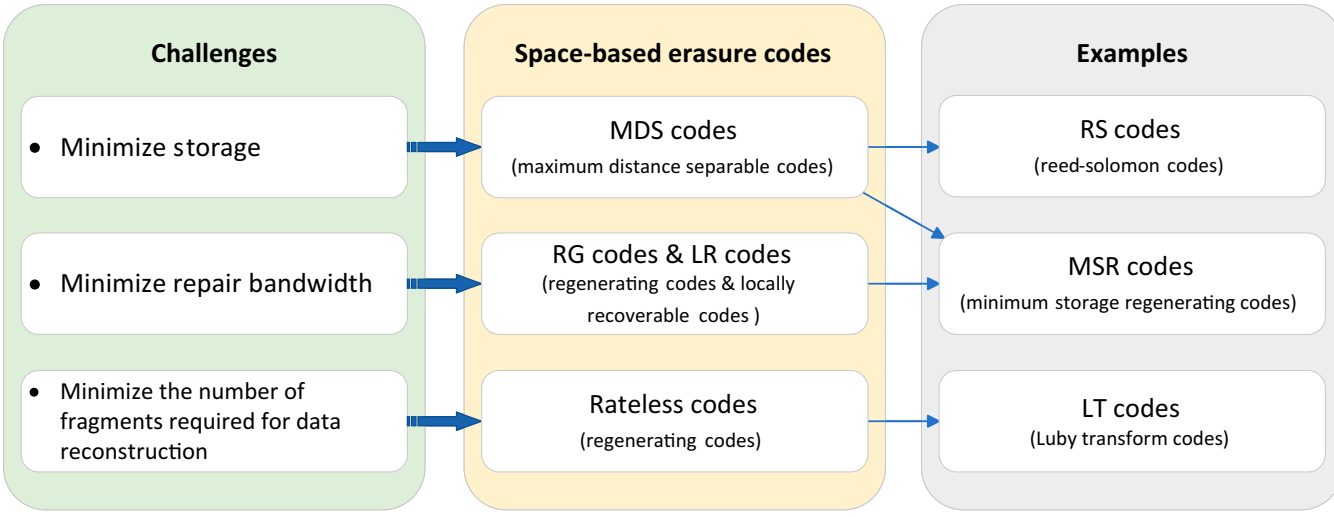

Fig. 5. An overview of erasure codes for distributed storage.

Wang et al. [10] proposed an adaptive minimum storage regeneration (AMSR) coding scheme, ensuring data reliability and efficient storage in satellite-ground IoT systems.

Additionally, to address the repair cost issues arising from link heterogeneity, Wang et al. [11] employed nonstacked repair strategies based on RG and generalized RG codes, optimizing data availability.

Minimization of the number of fragments required for data reconstruction

In satellite applications, satellite networks often face frequently changing topologies, unstable communication links, and unpredictable delays and packet loss rates. Therefore, a coding strategy that can flexibly respond to these challenges is needed. Rateless codes represent this strategy in erasure coding, as they can generate an unlimited number of encoded fragments, allowing the system to reconstruct data from any subset of these fragments. This flexibility makes rateless codes particularly suitable for scenarios with highly variable network conditions, greatly enhancing the efficiency and reliability of data reconstruction.

Advantages of erasure coding

By integrating these 3 optimization strategies, erasure codes can play a crucial role in enhancing data reliability, optimizing storage and bandwidth utilization, and improving system performance. Pang et al. [12] adopted MDS codes and rateless codes when migrating ground ultra-dense computing tasks to distributed satellite constellations, effectively improving system performance and optimizing latency and energy consumption. Additionally, the MDS and rateless coding strategies both address the issue of "stragglers" in the network (i.e., slower nodes), ensuring that data can be processed promptly and reliably even if delays occur in some parts.

Performance optimization: Caching and routing

Caching in space

As a widely used network performance optimization technique, caching can mitigate congestion caused by repeated service requests, reducing content retrieval latency and improving quality of experience (QoE) in satellite-ground integrated networks.

Satellite network caching storage strategies can be divided into 3 stages based on storage location: satellite-assisted caching, on-board caching, and caching in SAGIN.

Satellite-assisted caching

In the first stage, satellites are considered to provide backhaul transmission for content retrieving from the cloud, while content is only cached in terrestrial base stations (BSs).

Satellites can provide broader coverage and higher transmission rates in underserved areas or improve the efficiency of terrestrial networks.

In joint work sponsored by Princeton University and satellite service communication provider SES, Brinton et al. [13] were the first to use satellites to combat the explosion of data demands in CDNs. Instead of online caching in [13], Kalantari et al. [14] considered offline caching, where the content remains unchanged during delivery and is updated it during the subsequent placement phase. In addition, in their multibeam architecture, content placement can be tailored to the popularity of smaller regions, enhancing cache efficiency and hit rates over monobeam architecture. They further investigated satellite-aided edge caching systems in urban/rural areas and studied the effectiveness in mono/multi/hybrid-beam satellite mode [15]. These innovations aimed to reduce the time required for content placement while maintaining a high cache hit rate, thereby demonstrating the advantage of hybrid architectures in reducing latency.

On-board caching

With advanced on-board processing and larger storage capacity, satellites have grown considerably more powerful. Given their wide coverage and broadcast advantages, caching mobile data on satellites—especially for content distribution—is both practical and necessary.

This has led to the growing popularity of a hybrid caching model, where both satellites and ground stations store data. By proactively caching frequently requested content on edge satellites, popular data can be delivered directly to users, bypassing remote servers and backhaul links. This approach optimizes precious LEO satellite resources, such as link capacity and communication energy.

For example, Wu et al. [16] first presented a 2-layer caching model in satellite-terrestrial networks for content delivery services with ground station caches for the popular content in its local area and satellite caches for the most popular content in its whole coverage area. They conducted joint caching optimization to reduce satellite bandwidth use.

The genetic algorithm (GA)-based caching placement scheme in [16] is complex and converges slowly in large LEO networks. Therefore, Liu et al. [17] proposed a distributed caching algorithm using matching game theory to minimize access delay in LEO networks.

For integrated satellite/terrestrial radio access network (RAN), the combination of satellite and caching can both serve multiple small cells through broadcast transmission and provide efficient traffic offloading without introducing extra load on BSs nor getting pressure from the restriction of backhaul links. Li et al. [18] proposed a cooperative transmission scheme that introduced a cache-enabled LEO satellite network as part of the RAN. The cooperative transmission scheme achieved considerable improvements in traffic offload and energy efficiency and is suitable for handling greatly increased mobile traffic.

Han et al. [19] combined caching and multicast beam forming to propose a joint optimization problem. In summary, the long-term cache storage and the short-term content transmission problems were jointly considered to improve the cache utilization and spectrum efficiency in the STN.

Caching in SAGIN

Zhang et al. [20,21] employed deep reinforcement learning (DRL) to optimize caching placement and resource allocation in cache-enabled UAV nonorthogonal multiple access (NOMA) networks. They factored in the dynamics of UAV locations and the variability of content requests, thus filling a void in prior research, which predominantly centered on static environments. The traditional DRL is the single-agent algorithm, so it cannot handle an unstable environment when there are many agents in the scenario. When the number of agents increases, the unstable and dynamic environment will reduce the optimization performance. Li et al. [22] extended previous work by using multi-agent reinforcement learning (MARL). They proposed a NOMA-based framework for ground-satellite networks and introduced a method for deploying local cache pools, where

users, BSs, and satellites act as agents to cooperatively optimize resource allocation.

Routing in space

In wireless networks, traffic is dynamically routed across multiple paths to optimize performance. Satellite routing algorithms, crucial for SAGIN and STN, have been extensively studied.

To achieve effective data transmission based on satellite routing, it is necessary to consider the unbalanced distribution of global network traffic. In addition, the link connectivity or information accessibility among satellites is determined by the constellation topology and has very strong regularity. The shortest path, minimum delay, or maximum communication time goals are important considerations in solving satellite routing problems.

In static networks, the routing issue can be simplified to the shortest path problem and solved using Dijkstra's algorithm [23]. Wood et al. [24] adapted the open shortest path first (OSPF) protocol-based Internet Protocol (IP) routing to the dynamic satellite environment.

However, real-world satellite-terrestrial integrated networks have complex, dynamic topologies due to hierarchical structures and variable channel and traffic conditions (as shown in Fig. 2).

Routing based on virtual topology

Early routing protocols primarily focused on connection-oriented schemes, discretizing the time-varying satellite network into a sequence of fixed topologies (known as the virtual topology, VP, method) based on topological predictability. These methods are categorized into spatial and temporal virtualization.

Spatial virtualization algorithms include coverage area virtualization [25], which dynamically endows passing satellites with fixed logic addresses, and constellation network virtualization, which converts the dynamic network into static virtual nodes with a one-to-one correspondence to real satellites. Thus, the dynamic network is converted into static virtual nodes, turning the routing issue into a shortest path problem within a fixed topology. Space virtualization masks satellite mobility and offers robust adaptability. However, it demands a regularly shaped network, and route calculations based on local data are not optimal.

Temporal virtualization uses satellite periodicity to divide the network period into fixed time slices with invariant topology [26]. The shortest path algorithms is also applied within each slice [27]. The concept of snapshots [28] was introduced to describe the dynamics of LEO satellite networks, but this can lead to large overheads due to frequent inter-satellite link (ISL) changes. In [29], the snapshot concept was formalized to reduce storage overhead by storing only differences between adjacent snapshots. However, frequent routing computations are still unavoidable. Temporal virtualization requires accurate link status and optimal route calculations and thus is limited by low processing capacity due to offline computation. Moreover, many time slices require more memory.

Routing with optimization

Sigel et al. [30] proposed an ant colony-based optimization (ACO) framework. To address the local optimal solution issue in the ACO framework, Liu et al. [31] introduced a Kalman filter-based wolf colony optimization algorithm, where the Kalman filter is used to eliminate sudden traffic, while the wolf pack algorithm is used to determine the optimal path. Building on this work, Zhao et al. [32] improved the ACO framework to find the optimal set of links with multiple network constraints.

Routing with AI architectures

To simplify the intricate routing calculations, an AI model can analyze historical traffic data to forecast the ISL with the highest transmission capacity, thereby ensuring load balancing in the STN. Different deep learning architectures are proposed in the literature.

Reinforcement learning is well suited to dealing with sequential decision problems. Tu et al. [33] proposed a routing optimization method based on DRL for software-defined STN. Unlike OSPF, it adapts to changing flows and link status, improving end-to-end throughput and latency. Q-learning is a simple and efficient reinforcement learning method that has a fast convergence speed. Yin et al. [34] employed an accelerated Q-learning algorithm to identify the optimal routing strategy for STN.

To enhance training speed in the training process of ML, Na et al. [35] utilized an extreme learning machine (ELM) to predict upcoming traffic load at satellite nodes. In a supervised approach, architectures such as the fully convolutional neural network (FCNN) by Liu et al. [36] and CNN by Kato et al. [37] have been employed to solve routing problems.

AI's lack of interpretability can be problematic for users. For instance, while an AI might find an optimal path for video delivery, the path's duration might be inadequate. Fuzzy logic uses membership degrees to measure attributes, making it more flexible than Boolean logic. Considering that the CNN's judgment may contradict the user's QoE, Wang et al. [38] applied fuzzy logic to assess task requirements, enhancing the CNN's output for optimal path allocation.

Stochastic fault-tolerant routing

To address dynamic topologies and link instability in satellite networks, fault-tolerant routing algorithms must balance efficiency with robustness. Beyond traditional virtual topology and AI-optimized routing, the stochastic communication paradigm from network-on-chip (NoC) [39] offers promising alternatives. This model employs probabilistic broadcasting (e.g., nodes randomly select neighbor subsets for packet forwarding) to enable multipath redundancy. It tolerates data upsets, buffer overflows, and synchronization failures while maintaining low latency. This approach is particularly suitable for resource-constrained space environments where deterministic protocols struggle with intermittent connectivity.

Recent advances in NoC architectures demonstrate promising solutions for satellite multicast challenges. The cooperative network coding NoC (NCNoC) [40] employs corridor routing with orthogonal path splitting and adaptive flit dropping to overcome branch blocking while maintaining deadlock freedom. By enabling neighboring nodes to share encoding/decoding resources through cooperation units, it achieves a 127× multicast throughput improvement over conventional approaches with only 10.9% look-up table overhead. This lightweight, coding-aware routing approach is particularly suitable for satellite networks, as its path diversity naturally adapts to dynamic topologies while providing inherent resilience against cosmic ray-induced errors.

ICN-based satellite networks

With the development of satellite and information and communications technologies (ICTs), such as networking caching and computing, onboard caching in satellites has become a promising approach in practice. Many works consider how to leverage cache capacity to help optimize delivery, e.g., by building information-centric networking (ICN)-based satellite networks.

Satellite-assisted ICN

In-network caching can also be used to improve the performance of satellite-terrestrial networks. There exists some work on this issue. Siris et al. [41] proposed the first solution for integrating satellite and terrestrial networks within the ICN framework. This innovative approach includes leveraging the broadcasting and multicasting capabilities of satellites to refine content distribution and caching strategies. D'Oro et al. [42] presented SatCache, a pioneering caching strategy for information-centric satellite networks. This strategy cleverly employs the inherent broadcasting nature of satellite communications alongside user preference profiles to anticipate user interest in specific content types, thereby maximizing content hit rates and bolstering the resource efficiency of satellite communication systems. The offline content caching proposed by [14] is incorporated into hybrid satellite-terrestrial relay network (HSTRN) by An et al. [43] to alleviate the spectrum shortage and meet the requirements of improved spectral efficiency.

Jiang and Li [44] proposed a cooperative strategy to reduce satellite communication transmission delay, including a cache placement algorithm for downlink and a peer selection algorithm for uplink, thereby improving performance in terms of delay, hit rate, channel rate, and handoff rate.

ICN based on STN

As the scale of the satellite network increases, in-network caching can bring new vitality to intersatellite networking and routing. Many studies adopt a new network architecture that combines the ICN architecture with the STN.

In the perspective of networking, there is a growing consensus to expand SDN and network virtualization to STNs, which is referred as a software-defined and virtualized STN. Qiu et al. [45] modeled a joint resource allocation problem, encompassing caching, resource allocation, and computational resources, and addressed the problem using deep Q-learning methods.

Existing in-network caching schemes in satellite networks mostly acquire cache placement locations as well as cache content under static topological conditions. To tackle high-velocity node movements and fluctuating topologies, Xu et al. [46] proposed a hybrid caching approach for satellite networks. It initiates by stabilizing the dynamic process through timeslot segmentation based on similarities across layers. As nodes evolve in their spatial and temporal interactions, they are sorted dynamically.

Resilient edge-inspired storage for space systems

Important advances in terrestrial edge computing storage—addressing heterogeneity, resource constraints, and network instability—provide valuable insights for designing resilient storage systems adapted to space-specific challenges (high latency, intermittent connectivity, and strict resource limits). Key innovations from edge storage frameworks offer direct applicability to space systems:

Lightweight hybrid storage architecture: The edge storage component (ESC) in [47] demonstrates a lightweight hybrid architecture using containerization (Kubernetes K3s) and decentralized object storage (MinIO). Its dynamic lifecycle framework (DLF) enables transparent mounting of remote data sources, achieving 3- to 17-ms local operation latency on resource-constrained edge devices (e.g., Raspberry Pi).

For space systems, this suggests 3 key advantages: Containerized encapsulation resolves hardware heterogeneity across satellites and ground stations, while object storage combined with container storage interfaces abstracts storage for cross-platform deployment, and lightweight orchestration (e.g., K3s) minimizes operational overhead. The primary space adaptation challenges involve developing radiation-hardened container runtimes and optimizing DLF for volatile space-ground links.

Decentralized storage and fault tolerance: The fully decentralized system in [48] introduces dynamic sharding/replication distribution and topology-aware node discovery. Its algorithms autobalance storage loads during node churn, while segmentation throttling effectively mitigates distributed denial-of-service attacks.

For satellite networks, these mechanisms directly address dynamic challenges: Sharding and replication enhance data durability in constellations, ensuring survival during single-satellite failures; topology-aware discovery optimizes intersatellite routing by selecting low-latency or energy-sufficient neighbors; throttling prevents intersatellite link congestion. The critical challenge lies in scaling these LAN-based protocols to high-latency, disruption-prone space environments.

Standardized interfaces and heterogeneous platform integration: Analysis of edge platforms (AWS Greengrass, Azure IoT) in [49] highlights containerization (Docker) and standardized distributed storage (Ceph, HDFS) as foundational enablers for unified data access.

For space systems, this implies 3 strategic advantages: Unified onboard storage interfaces [e.g., S3-like application programming interfaces (APIs)] ensure cross-vendor interoperability; mature open-source solutions like Ceph accommodate diverse space data through native support for object/block/file storage; containerized decoupling considerably simplifies in-orbit updates and maintenance. The key requirement is developing lightweight protocols specifically adapted to space constraints including limited bandwidth and high latency.

Summary and lessons learned

In this section, we examined unique challenges and solutions for data storage and distribution in space computing environments. Unlike terrestrial systems, space-based storage must address the problems of intermittent connectivity, limited bandwidth, and harsh operating conditions while maintaining data integrity and accessibility. Key technologies include distributed storage architectures, erasure coding for reliability enhancement (e.g., MDS, regenerating, and rateless codes), and intelligent caching/routing strategies optimized for dynamic satellite networks. The integration of ICN further improves content delivery by leveraging in-network caching and name-based routing. Performance optimization techniques, such as AI-driven routing algorithms and adaptive caching placement, are critical to overcoming the constraints of space environments. From this review, we gather the following lessons:

- Resilience in harsh conditions: Space-based storage systems require specialized fault-tolerant designs (e.g., erasure coding) to handle intermittent links and hardware failures. Rateless codes excel in unpredictable network topologies, enabling flexible data reconstruction from partial fragments.
- Efficiency optimization: MDS codes minimize storage overhead, while regenerating codes reduce repair bandwidth during node failures. Caching strategies (e.g., hybrid satellite-terrestrial caching) must balance popularity awareness with storage constraints.
- Dynamic network adaptation: Traditional routing protocols fail in highly dynamic satellite networks; AI-based methods [e.g., DRL and ant colony optimization (ACO)] improve path selection under mobility. ICN architectures shift from host-centric to content-centric models, reducing latency but requiring updates for satellite mobility.
- Future directions: Space-adaptive hybrid storage policies will dynamically place data across onboard/intersatellite/ground tiers based on real-time link stability and energy levels; satellite swarm storage protocols must develop robust data distribution/retrieval methods for highly dynamic topologies with mobile satellites and intermittent links; finally, an integrated space-terrestrial storage platform should establish unified interfaces and management planes for seamless data flow across SAGIN infrastructures.

Virtualization and Scheduling of Space Computing Resources

Satellite resources, particularly computing power like graphics processing units (GPUs), are vital for various space applications, including remote sensing, navigation, and communication. Virtualization technology enhances resource efficiency and flexibility by transforming physical assets into configurable virtual resource pools. This allows for on-demand distribution and use of resources. Additionally, effective scheduling dynamically optimizes resource allocation based on mission requirements, resource status, and priority. This ensures that satellite systems can efficiently and reliably serve diverse users and application scenarios, maximizing the overall performance and utility of satellite resources in space.

Virtualization of computing resources

In aerospace environments, algorithm-driven data preprocessing on satellites is central to space sensing and space computing. The execution of on-board algorithmic models relies heavily on the support of GPU computing power. However, unlike ground-based systems, the computing power of space satellites cannot be easily replaced or upgraded, leading to a scarcity of computing resources. Therefore, it is essential to adopt GPU virtualization technology to improve the utilization of GPUs.

GPU virtualization is a technology that enables multiple VMs or containers to share one or more physical GPU resources [50]. In space computing scenarios, GPU virtualization can leverage technologies such as peripheral component interconnect (PCI) pass-through, hardware-assisted virtualization, API redirection, GPU full virtualization, and remote GPU virtualization. A comparison of typical GPU virtualization methods in space scenarios is given in Table 4.

Peripheral component interconnect pass-through

Applications deployed in space are often containerized or virtualized to enhance deployment flexibility. Initially, VMs and containers did not directly support GPU usage, which PCI pass-through addresses. As a basic yet effective GPU virtualization method, PCI pass-through assigns physical GPUs directly to VMs or containers, enabling them to control the allocated GPUs without additional overhead, thereby maximizing GPU performance [51]. In aerospace scenarios, this approach can allocate computing resources to high-priority applications to ensure efficient, stable, and independent operation.

The use of PCI pass-through for GPU virtualization in virtualized cloud computing environments was first proposed by Jo et al. [52]. Prior methods involved reimplementing GPU programming APIs and virtual device drivers at the level of the virtual machine monitor (VMM), leading to considerable performance overhead and maintenance complexity. In contrast, PCI pass-through directly bypasses the VMM layer, using the GPU's PCI express (PCI-E) channel, achieving transparency to GPU programming APIs with negligible overhead, comparable to bare-metal performance. Yang et al. [53] pointed out that

Table 4. Comparison of typical GPU virtualization methods in space scenarios

Property	Peripheral component interconnect pass-through	Hardware-assisted virtualization	Application programming interface redirection	Full virtualization	Remote GPU
Multi-user		✓	✓	✓	✓
Hardware-independent	✓		✓		✓
Near-native performance	✓	✓		✓	
High flexibility			✓	✓	✓
Low development cost	✓		✓		
Ease of maintenance	✓				
High stability	✓	✓		✓	

PCI pass-through technology enables VMs within the virtual environment to utilize Nvidia graphics cards, thereby allowing the utilization of Nvidia's Compute Unified Device Architecture (CUDA) for high-performance computing. Younge et al. [54] measured the performance of 2 Nvidia Tesla GPUs in Xen VMs using PCI pass-through and compared it to bare-metal hardware. The results showed minimal performance gaps, indicating that PCI pass-through in VMs is a viable option for many scientific computing workflows.

Hardware-assisted virtualization

For space computing, while high-priority tasks need dedicated GPU access, routine applications require efficient and flexible resource use. Thus, GPU virtualization—splitting a single GPU into shareable instances—is critical. PCI pass-through provides near-native performance but dedicates the entire GPU to one VM, hindering sharing and utilization [51]. Hardware-assisted GPU technology leverages input/output (I/O) virtualization hardware extensions provided by chipset manufacturers and GPU vendors to partition a single physical GPU into multiple independent (virtual) GPU devices. Each virtual GPU can be individually assigned and connected to a corresponding VM or container, enabling utilization through GPU device drivers within VMs or containers, thus supporting upper-layer applications [55].

Single-root I/O virtualization (SR-IOV) technology is a GPU hardware-assisted virtualization technique. SR-IOV achieves virtualization and multiplexing within the hardware, offering superior hardware control capabilities compared to software-based solutions [56]. By allocating dedicated I/O resources to virtual GPUs in this manner, it realizes resource isolation among VMs or containers, enhancing GPU resource utilization and flexibility without compromising performance [51].

In space computing, hardware-assisted virtualization provides excellent performance and environmental isolation [51]. Virtual GPUs can achieve near-native performance, making them ideal for computationally intensive and stable applications requiring high performance and reliability.

However, this approach necessitates coordination between underlying software and hardware, and implementing GPU scheduling policies based on it may be infeasible since GPU operations bypass the host operating system or hypervisor. Additionally, it lacks the real-time migration and fault-tolerant execution capabilities of API redirection methods [55].

API redirection

In the context of space computing power utilization, certain programs exhibit weaker requirements for isolation and stability, but prioritize flexibility in computing power. For such scenarios, the API redirection approach is suitable. API redirection overcomes the challenges of hardware dependency by modifying solely the GPU device software layer, without delving into driver or hardware-level specifics.

The core of API redirection involves establishing a wrapper library with the same API as the original GPU library. This wrapper library intercepts GPU calls from applications within the guest system, redirecting them through shared memory to the host operating system on the same machine or a remote machine with available GPUs. Upon completion of GPU processing on the host or remote side, the results are passed back to the application through the wrapper library [55]. The architecture of API redirection is shown in Fig. 6.

A key advantage of API redirection technology lies in its ability to virtualize GPUs and support GPU-utilizing applications without requiring underlying GPU driver implementation details.

However, maintaining the wrapper library necessitates updates whenever the GPU library of the vendor is updated [55]. In space environments, integrated hardware architecture GPU resources are frequently utilized, posing incompatibility

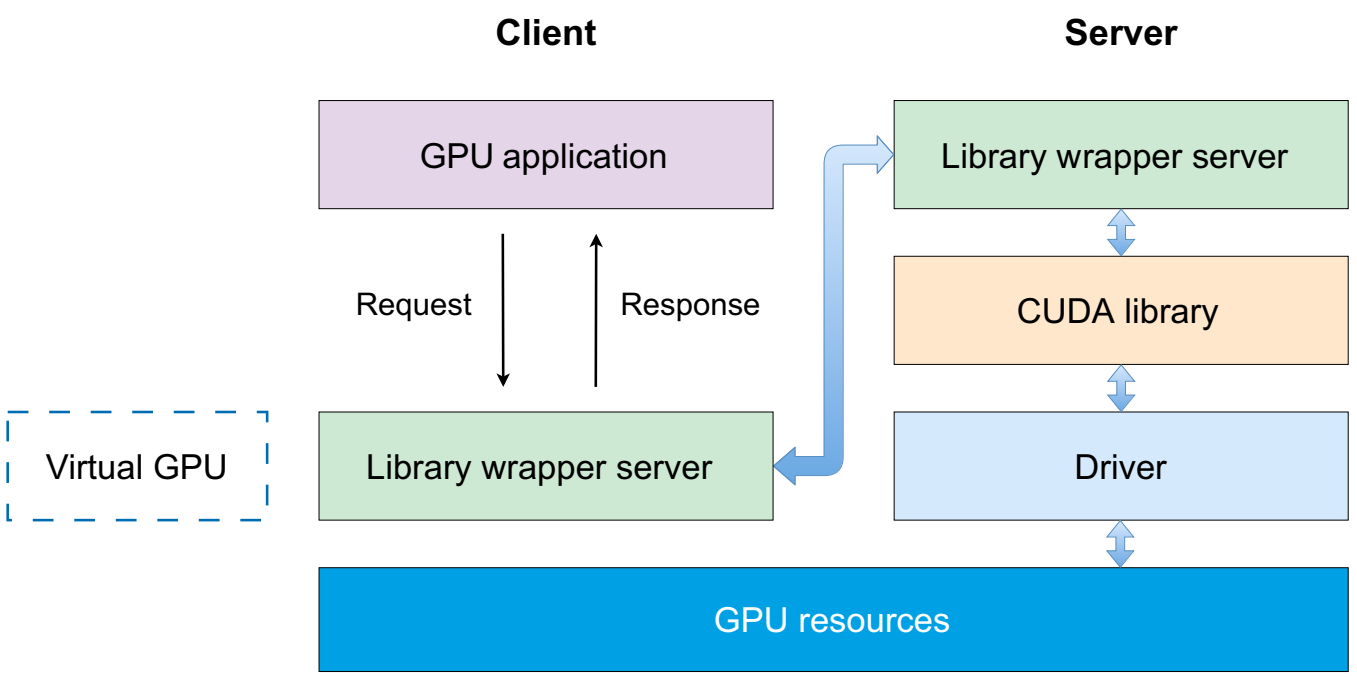

Fig. 6. Architecture of application programming interface redirection.

issues for conventional virtualization schemes due to their unique hardware architectures. Yang et al. [57] introduced sGPU, a method for virtualizing embedded GPU resources in edge computing environments. SGPU proposes a multi-container shared GPU algorithm to maximize the accuracy of computing power partitioning when the NVIDIA Management Library is are unavailable.

Full virtualization

Due to the high cost of replacing space resources, GPU full virtualization effectively balances the stability of hardware virtualization with the flexibility of API redirection. It allows for transparent utilization of customer VMs by integrating specific virtual GPU drivers within the VM's operating system. These drivers virtualize physical GPUs into independent virtual GPU devices, ensuring effective resource isolation, management, and application compatibility [51]. The architecture of full virtualization is shown in Fig. 7.

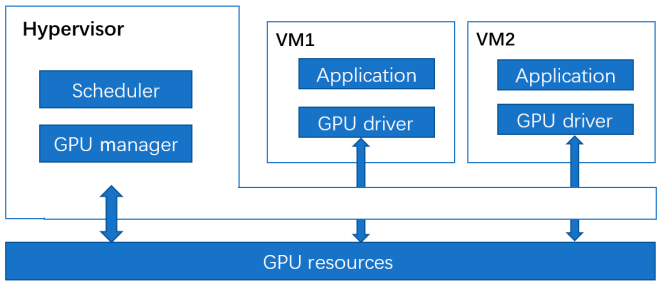

Fig. 7. The common architecture of full virtualization.

Remote GPU virtualization

In space application scenarios, the demand for remote calls of GPU computing power between smart satellite devices is relatively high. Remote calls of computing power fall within the scope of GPU virtualization technology, which relies on many implementation technologies, among which API redirection is the most common way. Remote GPU computing power resources can be utilized when isolation and stability requirements are low and flexibility requirements are high, and can effectively improve the utilization rate of the overall GPU computing power. In addition, remote calls to GPU virtualization technology can also save energy and reduce the resource overhead of the overall aerospace cluster equipment [58,59].

With the development of GPU virtualization technology, current mainstream technical frameworks supporting remote GPU virtualization all adopt a client–server distributed model structure. Typically, the client lacks local GPU resources or cannot directly access GPU resources, thus requiring remote calls to server-side GPU resources for application execution. The server node, usually equipped with one or multiple GPUs, provides remote GPU services to clients. The client–server architecture for remote GPU virtualization is illustrated in Fig. 8.

Orchestration and scheduling

Advancements in satellite network technology, especially the widespread deployment of LEO satellite constellations, have

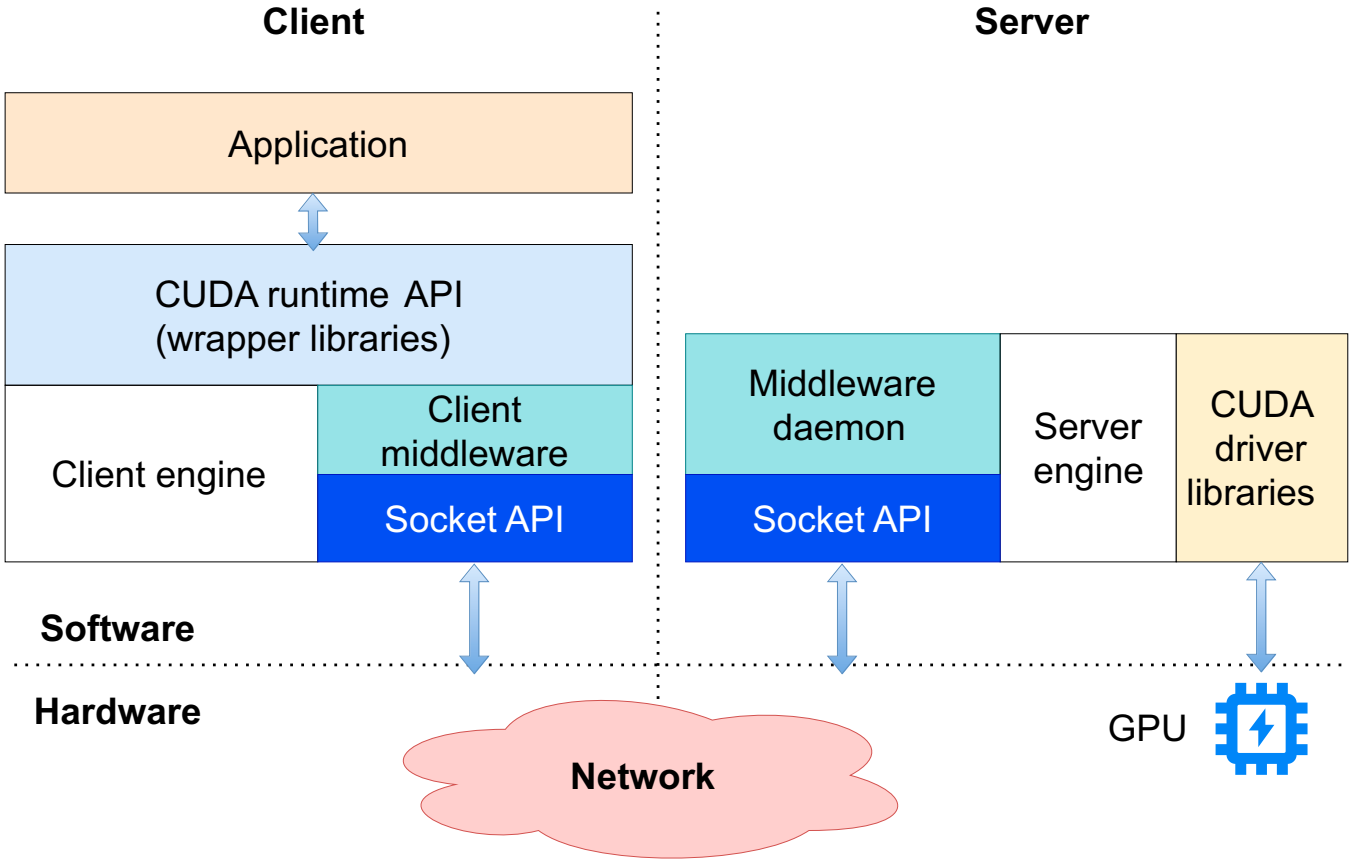

Fig. 8. Client–server architecture used in remote GPU virtualization.

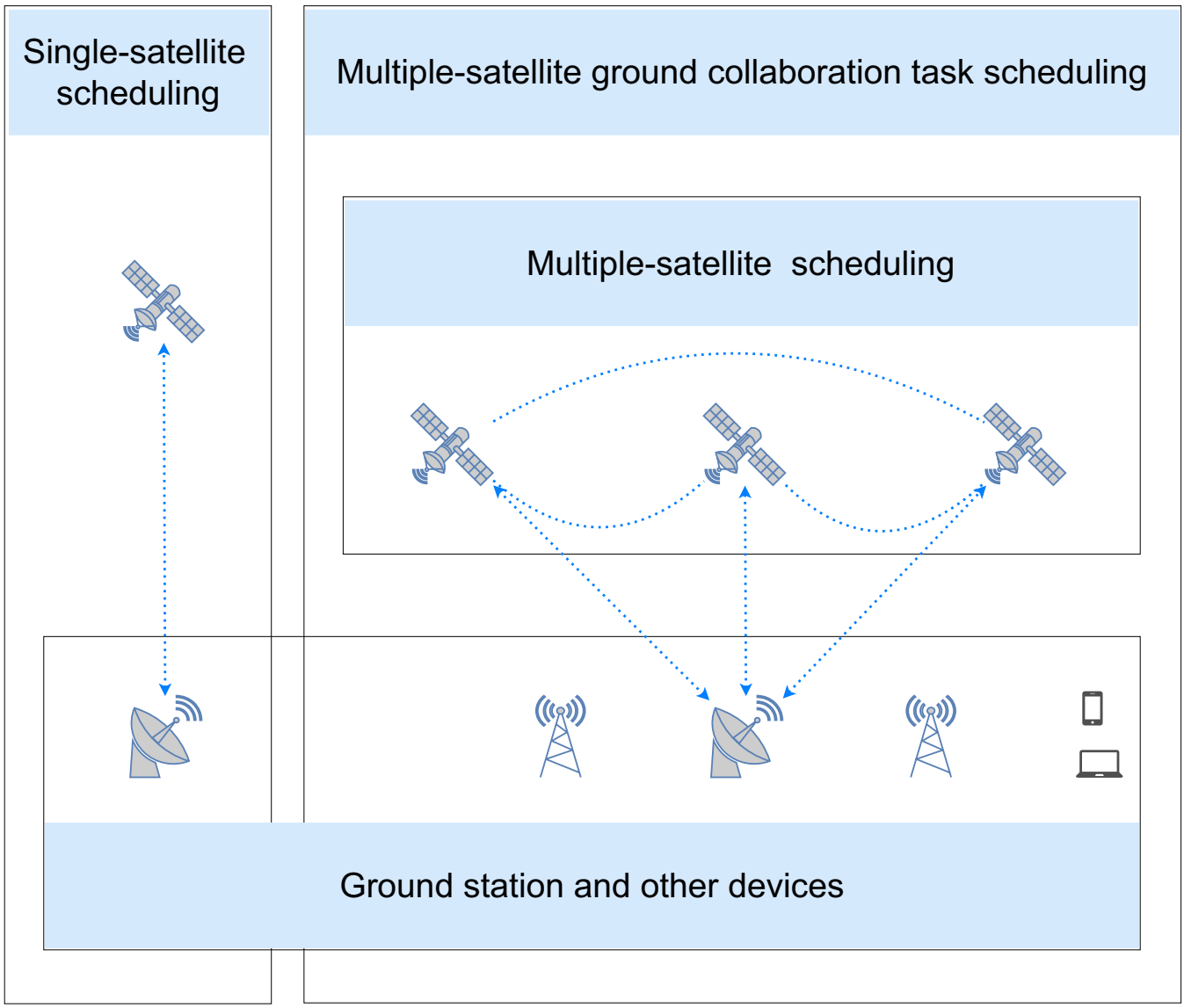

Fig. 9. An overview of satellite resource orchestration and scheduling.

introduced new challenges in the orchestration and scheduling of computing tasks. As satellite networks grow more complex, researchers are continually exploring innovative solutions to achieve efficient and intelligent task orchestration and scheduling and to ensure optimal use of network resources. As shown in Fig. 9, this survey summarizes current advanced technologies and future trends in 3 key areas: single-satellite, multi-satellite, and collaborative resource orchestration and scheduling for space-terrestrial networks. A summary of the main works is provided in Table 5.

Single-satellite resource orchestration and scheduling

Single-satellite resource scheduling pertains to the effective allocation and management of limited resources on a single satellite platform to ensure that the satellite can efficiently execute predetermined tasks. These resources typically include energy, storage space, communication bandwidth, and computing power. Single-satellite tasks encompass ground offline tasks and onboard online tasks, representing the earliest research direction in satellite resource scheduling. Examples of single-satellite scheduling systems include the automated scheduling and planning environment (ASPEN) system [60] aboard NASA's DS-1 and EO-1 satellites, and the advanced planning and scheduling initiative (APSI) system [61] utilized in European Space Agency (ESA) projects.

Carrel and Palmer [62] built upon the technologies of systems such as APSI and ASPEN by applying heuristic algorithms to scheduling systems. They designed the near-optimal evolutionary autonomous task-manager (NEAT) scheduling system, which uses a GA to maximize system throughput. However, NEAT focuses solely on scheduling and operational agents and lacks independent task priority ranking functionality. Consequently, it requires the integration of other algorithms to address issues such as resource conflicts, temporal dependencies, and task failures. To reduce task response time, Lu et al. [63] designed a single-satellite scheduling algorithm based on a greedy algorithm and a learning-based approach that could solve large-scale problems within seconds. However, it encounters inherent limitations in untrained areas and struggles to rectify inappropriate solutions. Zhang et al. [64] proposed an

Table 5. Summary of satellite resource orchestration and scheduling

Approache	Work	Key idea(s)	Trade-offs and shortcomings
Single-satellite resource scheduling and orchestration	[62]	Genetic algorithm-based NEAT scheme	Independent task priority lacking; integration required for conflict resolution
	[64]	Greedy algorithm-based OEC-TA	Data access and processing constraints for IoT devices
	[66]	Dynamic scheduling, SDN models, and AKG-based scheme for ECS	Memory consumption issue with BFS-based spanning tree
Multi-satellite resource scheduling and orchestration	[72]	AOC-based scheme	Insufficient theoretical support; central authority dependence
	[74]	Dynamic priority queue-based SDPLS	No consideration for remote area IoT device offloading
	[78]	SVM-based task allocation strategy selector combined with heuristic principles	Treats multi-satellite as cooperative agent; limited position and stability considerations
	[77]	Distributed cooperative dynamic task planning algorithm-based MARL	Increased problem-solving workload
	[79]	AMAS-based multi-satellite mission planning	Direct communication assumption (unrealistic in practice)
	[80]	Three-layer architecture and a distributed algorithm	Satellite movement impact unaccounted for; security concerns for offloading
Collaborative resource scheduling and orchestration for STNs	[82]	GAN-based algorithm	Spatiotemporal correlation with LEO satellites ignored
	[83]	Collaborative algorithm based on attention mechanism and proximal policy optimization	Dynamic grouping and aerial base station access unconsidered; unused cloud server power
	[84]	JCCRA-GM algorithm	Intersatellite cooperation not considered; suboptimal solutions from traditional methods
	[85]	SFC orchestration in satellite networks from game theory	Network dynamics and load balancing unaddressed
	[86]	Mathematical optimization and iterative thinking	Bandwidth assignment between edge and cloud offloading neglected
	[88]	Reconfigurable SAGIN architecture combining SDN and NFV	Limited to control functions; latency and throughput impacts unclear

orbital edge computing task allocation algorithm based on a greedy algorithm aiming to leverage satellite computing resources to provide services to ground users. It achieved better performance in terms of average delay and energy consumption compared to the double-edge computing algorithm. Nonetheless, it was unable to overcome the challenges posed by the lack of data access and processing resources in industrial IoT devices, which hindered the effective fulfillment of remote service quality requirements.

Zeleke and Kim [65] proposed a novel satellite autonomy strategy that combines CNN-based image classification technology with reinforcement learning-based task scheduling. Through simulation experiments, they demonstrated that the proposed algorithm, combined with reinforcement learning and Markov decision process algorithms, can considerably improve resource utilization efficiency. Additionally, Wang et al. [66] designed a single-satellite online resource scheduling algorithm for LEO edge computing satellites (ECSs) based on dynamic scheduling algorithms, SDN models, and the advanced *k*-means algorithm in ML. This algorithm effectively realized resource division for ECS and the construction of ISL. While the algorithm could directly implement continuous control of terminals based on dynamic adjustments, the breadth-first search-based spanning tree algorithm required substantial memory usage.

Multi-satellite resource orchestration and scheduling

Multi-satellite resource orchestration and scheduling involve task allocation, resource configuration, and scheduling in constellations of multiple satellites. Compared to single-satellite systems, multi-satellite systems typically exhibit higher observational efficiency, more comprehensive functionalities, and greater robustness. They offer distinct advantages in arranging routine observation tasks and responding to emergency tasks. Most multi-satellite task scheduling planning systems adopt either a centralized architecture [67–69] or a centralized-distributed architecture [70,71] and focus on task management and orchestration within the satellite constellation.

Iacopino et al. [72] proposed a system based on the ACO algorithm for multi-satellite mission planning and scheduling. Compared to GA-based systems, the ACO-based system demonstrated superior adaptability and coordination. However, the system lacked a solid theoretical foundation and struggled to ensure high-quality solutions without centralized authority. Tang et al. [73] established a 2-timescale hierarchical framework and proposed a heuristic-based atomic orbital search

approach to obtain superior policies with low computational complexity. Han et al. [74] introduced a satellite-based dynamic priority list scheduling (SDPLS) algorithm based on a dynamic priority queue. SDPLS addressed the issues of immediate response to time-sensitive tasks and efficient utilization of satellite cluster computing resources in highly dynamic edge clusters. However, the algorithm only considered offloading tasks from satellites to ground devices, neglecting the need for offloading computing tasks from remote IoT devices. In addition to the heuristic approach, Yang et al. [75] introduced a dynamic-distributed organizational structure and implemented an improved contract network protocol (ICNP) and a blackboard model negotiation mechanism to achieve balanced coordination in the multi-autonomous-satellite system. However, tasks were still processed sequentially in the ICNP, with each negotiation bidding for a single task, leading to increased communication volume as the scale of tasks grew.

Richards et al. [76] developed a software architecture for automated, distributed planning and coordination of satellite constellations in 2001, laying the groundwork for the application of ML algorithms in multi-satellite resource scheduling.

Chong et al. [77] proposed a distributed cooperative dynamic task planning algorithm based on MARL and introduced transfer learning into the MARL framework. This effectively addressed the challenge of large-scale dynamic task planning in multi-satellite systems. However, the MARL algorithm incurred high communication costs due to frequent interactions between agents and required a high level of intelligence across all satellites in the system. Yao et al. [78] developed a task allocation strategy selector combined with heuristic principles based on the support vector machine. It enabled the satellite constellation to autonomously select appropriate emergency task cooperative allocation strategies based on regular task information. However, the work treated the multi-satellite system as a cooperative agent without detailed consideration of communication instability and positional differences in the master satellite scenario.

Aside from the aforementioned heuristic algorithms and ML methods, several typical algorithms have been applied to multi-satellite resource orchestration and scheduling. For instance, Bonnet et al. [79] proposed a multi-satellite mission planning method using an adaptive multi-agent system (AMAS). The AMAS leveraged the self-adaptation and self-organization mechanisms to solve highly dynamic problems. However, the method assumed direct communication between all agents, which is unrealistic in practice. Tang et al. [80] converted the original nonconvex problem into a linear programming problem using binary variable relaxation and proposed a distributed algorithm based on the alternating direction method of multipliers to approximate the optimal solution with low computational complexity. However, the algorithm did not consider the impact of satellite motion on computation offloading and raised concerns regarding security during the offloading process. Lastly, Li et al. [81] addressed the collaborative management and scheduling of computing resources in 5G large-scale satellite networks. They proposed a satellite cooperative resource orchestration strategy (SCROS) utilizing NFV characteristics. SCROS refined the latency of each satellite involved in collaborative computation across different time phases, aiming to minimize the total. It achieved the lowest latency and highest computing resource utilization compared to other approaches to date.

Collaborative resource orchestration and scheduling for STNs

Space-terrestrial resource orchestration and scheduling technology are crucial for building an integrated space-ground information network. They aim to efficiently manage and allocate satellite resources, ensuring network effectiveness, reliability, and improved service quality, while facilitating successful mission execution. Research in this field incorporates advanced technologies such as intelligent algorithms, game theory, and mathematical optimization to advance the intelligence and optimization of resource scheduling.

He et al. [82] proposed the graph attention network (GAN)-based heuristic satellite orbit selection and resource allocation algorithm, a low-complexity method using GAN to handle graph-structured data. They also designed the load-aware service orchestration algorithm based on tabu search heuristics to solve service function chain (SFC) orchestration problems, achieving load balancing and high service acceptance with stable performance. However, these methods did not account for the spatio-temporal correlations of aerial base stations and LEO satellite networks. In [83], a collaborative algorithm based on the attention mechanism and proximal policy optimization was introduced, which modeled dependent tasks as directed acyclic graphs and used attention mechanisms to generate offloading decisions. However, these approaches failed to consider the dynamic grouping and access capabilities of aerial base stations, along with the computational power of cloud servers.

The concept of the Nash equilibrium in game theory facilitates fair resource distribution and ensures system efficiency and performance in space-terrestrial resource scheduling. Zhang et al. [84] developed the joint computation and communication resource allocation via game-theoretic matching (JCCRA-GM) algorithm, addressing resource allocation for computation-intensive services with low algorithmic complexity. However, the work did not consider collaborative computing among satellites, leading to suboptimal solutions. Qin et al. [85] in 2024 explored SFC orchestration in satellite networks using the best response, adaptive play, and stochastic learning algorithms from game theory, achieving higher resource utilization and lower service latency. However, this study did not consider dynamic network changes and load balancing, potentially leading to previous orchestration strategies no longer meeting user demands and rapid saturation of individual satellites, reducing equipment lifespan.

Mathematical optimization and modeling provide new insights into enhancing resource orchestration efficiency. Ding et al. [86] focused on resource optimization in the satellite-aerial integrated edge computing network (SAIECN), using mathematical optimization and iterative thinking to reduce energy consumption. However, the work did not systematically consider power, bandwidth, user association, and task scheduling, nor did it address bandwidth allocation between edge and cloud offloading. Li et al. [87] proposed the dynamic unified resource management algorithm, a dynamic task scheduling approach that monitors satellite resource usage and allocates tasks based on priority and availability. However, the work primarily focused on energy aspects and did not consider detailed resource allocation for storage and computing power, and did not conduct systematic analysis and testing of system throughput and response times. Additionally, He et al. [88] proposed a reconfigurable SAGIN architecture combining SDN and NFV. They formulated the resource allocation

problem as a mixed-integer nonlinear programming problem and used an iterative alternating optimization algorithm to find an approximate optimal solution. However, the LEO satellites were limited to control functions, and the impact of resource scheduling on latency and throughput was not thoroughly considered.

Summary and lessons learned

In this section, we explored the critical role of virtualization and scheduling in optimizing resource utilization for space computing systems. Virtualization technologies such as GPU pass-through, hardware-assisted virtualization, and API redirection enable efficient sharing of limited onboard resources (e.g., GPUs) across multiple tasks or VMs. Scheduling algorithms, ranging from heuristic methods to AI-driven approaches, dynamically allocate computational tasks across single satellites, constellations, or integrated space-terrestrial networks (e.g., STECN). Key challenges include balancing stability, flexibility, and performance in harsh space environments while addressing energy constraints and real-time processing demands. From the review, we gather the following lessons:

- Virtualization trade-offs: GPU pass-through offers near-native performance but lacks resource sharing. Hardware-assisted virtualization (e.g., SR-IOV) improves isolation and efficiency but requires specialized hardware. API redirection and full virtualization provide flexibility but introduce overhead in dynamic environments. Remote GPU calls enable resource pooling but face latency and compatibility challenges.
- Scheduling complexity: Single-satellite scheduling prioritizes task urgency (e.g., greedy algorithms) but struggles with scalability. Multi-satellite coordination leverages AI (e.g., MARL) and game theory for load balancing, although convergence speed and communication costs remain hurdles. Cross-domain orchestration (e.g., SAGIN) demands hybrid SDN/NFV architectures but must account for latency and security in heterogeneous networks.
- Key insight: Virtualization and scheduling are pivotal for scalable, resilient space computing, but their success hinges on tailoring solutions to the unique constraints of orbital environments—limited power, intermittent connectivity, and mission-critical reliability. Future work must bridge the gap between theoretical models and deployable systems, emphasizing lightweight, adaptive technologies.

AI for Space-Based Distributed Computing

Recent advances in AI for aerospace applications have brought about new opportunities for fast-growing satellite and space computing. The ESA launched the first artificially intelligent European Earth observation mission in September 2020 [89].

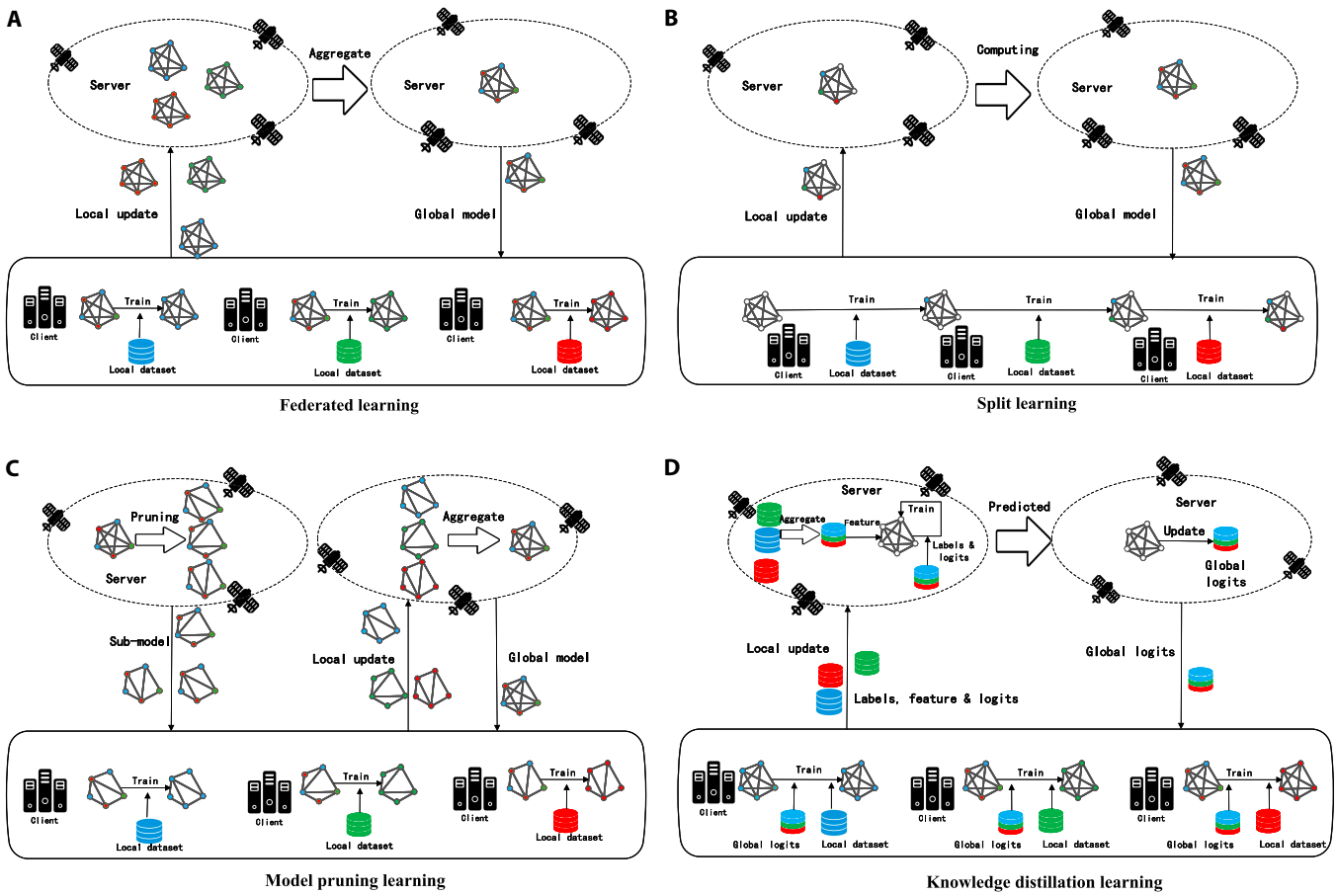

Fig. 10. Schematic diagram of distributed collaborative learning in space computing. (A) Federated learning. (B) Split learning. (C) Model pruning learning. (D) Knowledge distillation learning.

In the space environment, various computing satellites and ground computing centers serve as the primary carriers of computation. As shown in Fig. 10, AI-based distributed collaborative computing in space addresses the limitations of single-point computation, ensuring data security and privacy while enabling efficient data processing and model co-optimization through collaboration among multiple satellites and between satellites and ground stations. While research in space collaborative computing is still in a nascent stage, a number of interesting studies have recently appeared, of which we will highlight some that represent noteworthy progress.

Distributed collaborative learning

The distributed collaborative learning of space computing systems offers numerous benefits, including enhanced data security, increased system reliability, improved computational efficiency, facilitated data sharing, reduced costs, and enhanced user experience. This architecture is particularly well suited for applications that require handling large volumes of geospatial data and real-time interactions.

Federated learning

Since the introduction of the classic federated learning (FL) method by McMahan et al. [90], FL has emerged as a widely adopted distributed ML paradigm. This paradigm enables multiple nodes to train local models using their local data and upload model parameters to a central server for global model aggregation, thereby achieving collaborative model training across nodes. However, traditional FL frameworks often assume ideal communication conditions between nodes, with no constraints on communication bandwidth and time. In space computing scenarios, the performance of FL faces considerable challenges due to limited downlink bandwidth and brief communication windows. Decentralization presents one solution to these challenges. Zhai et al. [91] designed a decentralized FL framework that enables efficient model aggregation in LEO satellite networks without a central server, thus circumventing the bandwidth limitations associated with central servers in FL. Similarly, Zhang et al. [92] proposed satellite-edge leader FL architecture (SELFL), a ground station-independent satellite distributed FL architecture. This architecture facilitates the transmission of local model parameters and the aggregation and updating of global models between selected leader satellites and follower satellites, based on established intersatellite links, by evaluating satellite connectivity and load dynamics. Furthermore, it incorporates a novel DRL-based resource allocation strategy that employs distributed proximal policy optimization (DPPO) to optimize both the computational capacity and transmission power of satellites, ultimately enhancing FL efficiency while minimizing energy expenditure.

Addressing the issue of efficient resource allocation in LEO satellite networks, Han et al. [93] proposed a training delay minimizer that optimizes satellite-specific network resources, including the amount of data offloaded from ground devices to the satellite and the computational speed of the satellite. In order to achieve concurrent transmission of large FL models in a short period of time and sporadic visible windows, Elmahallawy et al. [94] proposed a new FL-SatCom framework to use orthogonal frequency division multiplexing (OFDM) for communication and a new FL model aggregation scheme to avoid a biased global model.

Split learning

Split learning in space computing enables decentralized model training by allowing data to remain on local devices while sharing only intermediate representations, thereby enhancing privacy and reducing communication overhead. Lin et al. [95] proposed efficient parallel split learning (EPSL) to accelerate model training. EPSL parallelizes client model training and reduces the dimension of the activation gradient of the backward propagation (BP) through the last layer gradient aggregation, considerably reducing the server-side training and communication delay. By considering the heterogeneous channel conditions and computing capabilities of edge devices, this method jointly optimizes subchannel allocation, power control, and cutting layer selection to minimize the delay per round. To enhance the efficiency of collaborative learning between the ground and space, Jiang et al. [96] proposed a privacy-preserving general federated split learning long short-term memory (LSTM) model for processing sequential data in the integrated satellite-terrestrial network, where the ground uses split learning methods, and mobile edge servers send their locally trained subnetworks to the satellite server to aggregate into a global model.

Lin et al. [97] explored the combination of split edge learning (SEL) and MEC to achieve decentralized data processing, reduce the cost of backhaul bandwidth, achieve ultra-low latency, and enhance context awareness. The article proposed a hierarchical split learning architecture that supports collaborative training of large-scale models, optimizes resource usage efficiency through dynamic resource allocation strategies, and builds a network of edge servers through multi-hop split learning to share the workload of computationally intensive model training. In the context of task splitting for deep neural network (DNN) computations in multi-satellite systems, Peng et al. [98] proposed a collaborative satellite computing workload balancing adaptive task partitioning scheme. This scheme dynamically divides DNN tasks into multiple subtask blocks and employs a binary search method to determine the maximum acceptable workload for each block, ensuring that the partitioned task segments can be evenly distributed among satellites, thereby enhancing the utilization of satellite computing resources. Regarding collaborative inference in multi-satellite DNN networks, Guan et al. [99] introduced a dynamic multi-satellite inference method that selects partitioning points. This method splits the inference task into 2 parts: The task owner executes the first part, and the remaining part is allocated to other task performers. By employing a task gain-aware approach, it balances the completion and accuracy of the task.

Model compression and adaptation

Owing to the limitations in terms of the size and resources of satellites, their computing and storage capacities are restricted and fail to meet the computing and storage demands of complex models, especially those of larger-scale models. Consequently, it is essential to compress the size of models in a manner commensurate with the resources and computing capabilities of satellites while keeping the performance loss within an acceptable range.

Model pruning

Model pruning is a model compression method applicable to distributed ML. It reduces the scale of the model by pruning the network structure of the model, simultaneously enhancing

the speed of the neural network and minimizing its accuracy loss. Sun et al. [100] combined differential privacy with graph and model pruning and proposed the dynamic topology-informed pruning (DTIP) method to optimize graph neural networks (GNNs) for distributed learning. DTIP applies differential privacy to the original graph data and prunes the GNNs, and stores a portion of the model at the space station and the ground station, respectively, for split learning, thereby optimizing the model size and the communication load across network layers.

Jiang et al. [96] combined model pruning and split learning and introduced the LSTM model to handle sequential data in the satellite-ground network. The LSTM model is pruned and split to reduce the computational workload and the consumption of communication resources. Each ground station owns the pruned submodel of the LSTM and conducts split learning for model training. Subsequently, the locally trained submodels are sent to the satellite and all the received submodels are aggregated into a global model. Lin et al. [101] put forward a general FL framework for LEO satellite networks, named FedSN in response to 3 challenges, namely, the heterogeneous computing and memory capabilities of different satellites, the limited downlink/uplink rates, and model staleness. It treats all satellites visible to the ground station at different contact times as an intergroup set, and groups satellites in different orbital planes during each contact as an intra-group set. For every intra-group set, it then assigns the optimal subset of pruned model components to the satellites within that group.

After model training, all the satellite models in the intra-group set are aggregated into basic submodels for subsequent global model aggregation. Finally, a pseudo-synchronous model aggregation strategy is adopted to aggregate the global models formed by the satellites in different intergroup sets into one global model.

Knowledge distillation

Knowledge distillation enables a small model to learn the knowledge output by a larger-scale teacher model during the training process of the small model so that the small model can also achieve higher performance or even performance close to that of the large model. While realizing model compression, it also allows for heterogeneity among models of different clients in distributed learning and improves the flexibility of distributed learning. In order to compress large deep learning models and deploy them on satellites with limited resources to perform tasks, Wang et al. [102] proposed an onboard change detection method based on knowledge distillation, which includes prototypical contrastive (PC) distillation and channel-spatial-normalized (CSN) distillation. PC distillation improves the detection ability of student models in changing areas with similar features to the background. CSN distillation guides the student model to accurately identify the changing areas with complex shapes. Pang et al. [103] proposed a pyramid distillation framework, which stacks multiple groups of deep mutual learning models, with smaller models placed on top of larger models. It can automatically and flexibly adjust the number of pyramid layers to obtain models of different accuracy corresponding to different compression ratios (CRs), thereby obtaining the influence of model CR on the law of model accuracy change.

Elmahallawy and Luo [104] proposed the one-shot FL method LEOShot in response to the delay caused by multiple rounds of communication between satellites and ground stations in FL. The entire training process can be completed with only a single round of communication, and it allows for heterogeneity among models of satellites and ground stations. LEOShot constructs a generator to generate high-quality unlabeled synthetic data and divides these data into multiple groups with the same distribution. Then, on each data group, the models of all satellites are used as teachers, and a model copy is trained by calculating the average logits of all satellite models as the output of the teacher model. Finally, all model copies are aggregated to generate the final global model.

Summary and lessons learned

In this section, we reviewed the main AI methods for space-based distributed computing. The methods are summarized in Table 6, and the lessons learned are as follows:

- AI-based space-based distributed computing involves the integration of AI techniques with distributed computing systems that operate in space environments, such as satellite networks. This field aims to leverage the capabilities of AI to enhance the efficiency, autonomy, and intelligence of space-based computing systems, enabling them to better handle complex tasks and data processing in orbit.
- The field of AI-based space-based distributed computing is advancing rapidly, with key focus areas including FL, which trains AI models across satellites and ground stations without centralizing data, addressing bandwidth and connectivity challenges. Split learning enhances this by dividing the training process among nodes, preserving privacy and optimizing resource use in constrained environments. Additionally, model compression techniques, such as pruning and knowledge distillation, reduce AI model sizes for deployment on satellites while maintaining performance.

Security and Privacy of Space Computing

The security and privacy of space computing systems are critically important due to the unique challenges these systems face. Node exposure, open channels, and highly dynamic topologies make satellite networks vulnerable to various cyber threats, such as eavesdropping, tampering, and denial-of-service attacks. These vulnerabilities can compromise sensitive information and critical infrastructure, affecting national security and economic interests. To address these risks, advanced security measures, including modern cryptography, blockchain, and quantum communication, are being extensively researched. This section will provide a comprehensive review, and a summary of the main works is provided in Table 7.

Encryption for confidentiality

Encryption algorithms

Satellite signals are susceptible to various forms of interference and eavesdropping as they traverse the atmosphere and space. Encryption technology plays a crucial role in safeguarding the security of information transmission in satellite networks.

Shortly after the concept of public key encryption was introduced, Ingemarsson and Wong [105] incorporated this idea into the design of satellite communication systems. They

Table 6. Summary of AI approaches for space-based distributed computing

Approache	Work	Key idea(s)	Trade-offs and shortcomings
Federated learning	[91]	Designs a decentralized FL framework supporting various resource-intensive training tasks	Heterogeneity will bring high computational complexity
	[93]	Proposes a ground-to-satellite collaborative FL method	Satellites connected in specific areas affect data transmission
	[94]	Supports the simultaneous transmission of large FL models within short and sporadic visible windows	High computing and communication resource requirements
Split learning	[95]	Parallelizes client model training and reduces the dimension of the activation gradient	Requires higher resources for model segmentation
	[96]	Proposes a privacy-preserving general split-federal learning model	Communication links may have bandwidth limitations
Model pruning learning	[100]	Combines differential privacy with graph pruning and model pruning to optimize GNNs in distributed learning and conduct pruning on GNNs	Differential privacy necessitates a higher computational overhead
	[96]	Combines model pruning with split learning for effective pruning and segmentation of the global model	Only the LSTM model for sequential data processing is considered, limiting the flexibility of model selection
	[101]	Partitions satellites into inter- and intra-group sets, performs pruning and allocation within intra-groups, and aggregates models globally across intergroups	Constrained by limited intersatellite bandwidth, a pseudo-synchronous aggregation strategy is adopted
Knowledge distillation learning	[102]	Optimizes the satellite model through prototypical contrastive (PC) distillation and channel-spatial-normalized (CSN) distillation	The communication limitations between satellites and ground stations have not been taken into account
	[104]	Proposes a one-shot FL method, the approach accommodates the heterogeneity between satellite and ground station models	The communication constraints among satellites may potentially affect the efficiency of training model replicas

outlined 2 approaches for key distribution and authentication utilizing the onboard processing capabilities of satellites. One method involved an identifier to facilitate the distribution of encryption keys, and the other utilized a one-way function to address the issues of key distribution and authentication. Cruickshank [106] took a more comprehensive view, exploring the application of public key and symmetric key encryption techniques in satellite networks. He proposed a security system combining public and secret key systems designed for personal satellite communication systems. His scheme allowed the satellite network to transmit information without any security processing, facilitated seamless integration with existing wired networks, and met the security needs of commercial organizations.

Image data constitute a major type of data acquired and stored by satellites. Due to the massive size of satellite image datasets, image encryption algorithms must exhibit high efficiency, strong diffusion and confusion abilities, and resistance to brute-force attacks. Researchers have refined traditional symmetric encryption algorithms to enhance the efficiency of image data encryption. A study by Ortakci and Abdullah [107] compares the performance of the advanced encryption standard (AES) and 3DES algorithms in image encryption and concludes that AES considerably outperformed 3DES in computational time. Ning et al. [108] combined the AES and low-density parity-check (LDPC) codes to design a secure and reliable error control method called SEEC. Simulation results in MATLAB showed that the SEEC method could accurately recover original data at signal-to-noise ratios (SNRs) equal to or greater than 2.5 dB, demonstrating excellent error correction and encryption performance.

To further enhance the efficiency and security of confidential satellite communications, Murtaza et al. [109] developed an efficient symmetric encryption algorithm specifically for satellite communications. This algorithm adopted the concept of perfect forward secrecy (PFS), generating message keys through matrix iteration. Experimental results demonstrated that in hardware implementation, a single clock cycle was sufficient to perform the exclusive OR operation and generate ciphertext. However, the limitations of relying solely on a single encryption method were highlighted by Jeon et al. [110], who pointed out that integrating encryption with beamforming

Table 7. Summary of security approaches for satellite networks

Approache	Work	Key idea(s)	Trade-offs and shortcomings
Encryption algorithms	[105,106]	Hybrid encryption for satellite communication	Limited to 2 parties; multi-party participation may incur high communication overhead
	[107,108]	Efficient symmetric encryption algorithms for image data	Standalone module; lack joint optimization with image processing algorithms
	[109,110]	New protocols for satellite communication, including PFS symmetric algorithm and integrating encryption with beamforming technology	Lack empirical validation; research potential remains
	[111]	PCMAC algorithm with AES and parallel CMAC	Improved performance; additional computational resources
	[112]	Enhance data security in critical satellite systems by ensuring data authentication, integrity, and confidentiality	Specific standard; ongoing challenges in satellite communication
Authentication	[115]	Consortium blockchain-based authentication using replica storage nodes	Complex setup, constant identity updates
	[116]	Handover authentication through batch verifications using biometrics, ECC, and digital signatures	Enhanced performance; biometric reliance, high implementation cost
	[121]	Blockchain's decentralized nature and id-based crypto for access and handover authentication	Centralized private key generator reliance results in vulnerability and complexity
	[122]	Lattice-based authentication scheme for satellite communication	Effective IoT management; implementation complexity, resource demands
	[124]	Cross-domain authentication system: decentralized architecture, NIZK proofs, etc.	Integration complexity, efficiency
Key management	[114]	Blockchain-based distributed key management system using BLS and cross-domain trust	Increased complexity, crucial security aspects
	[126–128]	QKD-based key management (satellite-assisted KEMs, constellation modeling, multi-layer quantum satellite networks)	High costs and instability for quantum computing devices
	[129–132]	Lightweight key agreement (ID-based, OTP-based, hash-based)	Two-party focus; new attack vulnerability
	[133–137]	Lattice-based post-quantum key agreement protocols	Nonstandardized algorithms used; potential attack vulnerabilities; lower efficiency

technology could provide more comprehensive protection for space-terrestrial integrated networks, particularly addressing security issues arising from data explosions in B5G/6G networks. They overcame the drawbacks of using either encryption or physical layer security alone, effectively enhancing the security of satellite networks through the synergistic use of encryption and beamforming technologies. Future applications hold considerable potential in leveraging big data technologies to improve the efficiency of encrypting large volumes of satellite images and to accommodate real-time and large-scale data transmission demand.

Authenticated encryption

An effective method for protecting data security in satellite communication is the use of authenticated encryption (AE) [111,112]. AE combines encryption and authentication to ensure data integrity, confidentiality, and authenticity. This dual protection prevents tampering and eavesdropping, ensuring that transmitted data remain unaltered and confidential throughout its journey. The integration of encryption and data authentication is particularly advantageous for satellite systems with limited computing resources, optimizing their performance. An early AE algorithm with a parallel architecture, called parallel cipher-based message authentication code (PCMAC), was introduced by Hussain et al. [111]. This algorithm leverages the AES algorithm and a parallel implementation of the CMAC authentication algorithm to enhance performance. More recently, Tawfik et al. [112] proposed an AE approach that aimed to optimize data security in critical satellite systems and mitigating risks associated with current military standards by ensuring data authentication, integrity, and confidentiality. However, their work is limited to a specific standard. Despite the advancements in AE for data security, its application within satellite networks remains in a nascent stage. As the reliance on satellite networks continues to grow, advancing AE techniques to meet the specific needs of space-based systems will be critical for the security and resilience of future space infrastructures.

Research on the security of data for space computing is still in its early phase, with various open challenges that must be addressed to balance security and efficiency within the unique structure of satellite networks. As space computing continues to develop, it is imperative that security strategies evolve to address the complexities of decentralized, dynamic, and privacy-sensitive satellite networks, ensuring the long-term sustainability of these critical systems.

Authentication

The need for authentication mechanisms is increasingly vital to ensure the integrity and confidentiality of transmitted data between satellites and mobile devices. Among the security challenges to consider are dynamic network formation, identity privacy, and handover authentication [113]. Moreover, due to the fast development of the satellite industry, cross-domain authentication is critical for handover communication [114]. Authentication mechanisms can be classified into satellite constellation authentication and mobile node authentication.

Satellite constellation authentication

The fast development of satellite constellations requires efficient and scalable authentication solutions. In 2022, Xiong et al. [115] presented a consortium blockchain-based and privacy-preserving authentication scheme for satellite networks. It aims to enhance security in interconstellation collaboration within space-ground integrated networks. Their approach uses replica storage nodes (RSNs) to cache the latest blocks and provide query service in order to reduce query delays. While innovative, this scheme requires a ground-based identity management and blockchain server (IMBS) that continuously updates blocks across all RSNs, increasing system complexity and management overhead. Additionally, the trade-off between dynamicity, decentralization, security, and efficiency remains an unresolved challenge for satellite authentication in space computing.

Mobile node authentication

Authentication is particularly critical in the context of terrestrial mobile device-satellite communication, where mobile devices switch their communication links between different satellites. Moreover, the balance between privacy and access control for cross-platform satellite communications must be considered.

Access and handover authentication

As mobile devices move across different satellite coverage areas, they must seamlessly authenticate and transfer their connections to new satellites without compromising security. In the last few years, several approaches have been proposed to solve this problem [114,116–119]. In 2020, Xue et al. [117] proposed a lightweight handover authentication method based on group key agreement and key agreement in SDN settings. The disadvantage of this method is that it does not provide anonymity; therefore, private information is disclosed. Recently, a more efficient group access-based handover authentication approach that considers the dynamicity of 5G networks focusing on high-speed rail communication was presented by Yang et al. [119]. The security of this approach relies on the Chinese remainder theorem. However, it only considers the specific field of railways and not other networks. Recently, Guo et al. [116] improved the performance of handover authentication using batch verifications using biometrics from the user, elliptic curve cryptography (ECC) and digital signatures of the ISO/ICE 14888-3 SM2 standard, but the security relies on biometrics, which is complex and costly in terms of implementation. Li et al. [120] and Guan et al. [121] have exploited the decentralized characteristic of the blockchain to propose an access and handover authentication approach using identity-based cryptography. However, blockchain-based approaches limit scalability and increase complexity. Ma et al. [122] proposed a lattice-based authentication considering the massive number of IoT devices in satellite communication. It provides mutual authentication, unlinkability, and data confidentiality, and it is resistant to quantum attacks. Yang et al. [123] solved the anonymity problem of authentication using group signatures, but their method relies on a centralized and trusted credential manager.

Cross-domain authentication

Cross-domain authentication is essential for ensuring secure and uninterrupted access across multiple satellite network domains and space computing environments. To address this problem, in 2021, Liu et al. [124] proposed cross-domain authentication with a decentralized authentication architecture to solve the fair billing problem and thus ensure that users and service providers are treated equitably. It is based on non-interactive zero-knowledge (NIZK) proofs and smart contracts with ring signatures. However, it has high computational costs. In 2023, Liu et al. [125] solved the cross-domain handover authentication problem using a hierarchical structure of space, air, and terrestrial networks, where satellites belong to groups according to their orbital heights and are managed by a group leader. Recently, Wang et al. [114] proposed a blockchain-based distributed key management method that facilitates trust and parameter sharing across different domains. Their approach employs a combination of Boneh–Lynn–Shacham (BLS) signatures and group signatures to enable batch anonymous authentication. Despite the increased complexity of the proposed approach, this work addresses many crucial security aspects for cross-domain communication.

Challenges and future work

In summary, ensuring handover authentication that considers real-time communication and adapting to dynamic environments are open issues that need to be addressed. Future work and open problems include developing scalable solutions, improving resource efficiency, and refining trust management mechanisms to support the growing complexity of satellite networks.

Key management

Effective key management is fundamental to ensuring the security and reliability of satellite network communication. Given the vast distances involved, the dynamic nature of satellite orbits, and the potential for considerable delays, traditional key management techniques often fall short of meeting the stringent requirements of satellite systems. Key management in satellite networks must address several critical challenges, including secure key distribution, frequent key updates, and the ability to handle large numbers of terminal nodes efficiently.

Quantum key distribution

In satellite communication security, quantum key distribution (QKD) marks a paradigm shift with its robust cryptographic

capabilities. Satellite-assisted QKD-based key encapsulation mechanisms (KEMs) [126] integrate high-security quantum AES-256-bit keys with classical communication speeds, enabling long-distance key transmission and supporting numerous terminal nodes. Trusted-node QKD networks using satellite constellations [127] and combining LEO with GEO satellites improve key establishment between ground stations, addressing latency issues and fostering global quantum-safe networks. QKD over double-layer quantum satellite networks [128] integrates GEO and LEO satellites for enhanced key relay success rates. Simulations explore the effects of routing, link numbers, node capabilities, and service granularity on network performance, guiding QKD optimization. In summary, through satellite-assisted KEMs [126], constellation modeling [127], and multi-layer quantum satellite networks [128], researchers are overcoming obstacles to global QKD deployment. These developments strengthen satellite communication security and pave the way for quantum-safe networks, essential in the era of quantum computing.

Lightweight key agreement

Satellite communication systems often struggle with limited computational and storage capacities, along with unpredictable communication environments. Traditional security protocols have struggled to keep up with demands for processing speed, bandwidth efficiency, and robustness against various threats. To address these challenges, lightweight key agreement mechanisms have been developed to enhance security and operational performance. In 2010, Yantao and Jianfeng [129] introduced an identity-based cryptographic framework that integrates hash values into the Diffie–Hellman protocol, improving communication and computational efficiency in satellite networks. However, this scheme might still be susceptible to specific types of attacks, such as those exploiting weaknesses in the hash function used. Building upon this foundation, Song and Lee [130] refined the key exchange protocol with timestamping mechanisms to secure communications between the network control center and the return channel satellite terminal. While their approach effectively mitigated man-in-the-middle attacks, it could potentially introduce additional latency due to the use of timestamps. To further address processing speed and bandwidth usage issues, in 2020, Altaf et al. [131] developed a protocol featuring strong mutual authentication and session key sharing using public key infrastructure and XOR operations. Although this strategy optimized computational complexity and speed, it may require more frequent key updates, which could increase overhead in resource-constrained environments. In the same year, Murtaza et al. [132] focused on the specific requirements of LEO satellite communication systems. They devised a protocol with random numbers and hash-based authentication that withstands various attacks while maintaining low overheads. However, the reliance on random numbers could make the system vulnerable if the random number generator is compromised. These advancements collectively aim to strike a balance between security and efficiency in satellite communication environments while recognizing the need for ongoing refinement to address emerging threats and technological limitations.

Post-quantum protocols

The series of studies on post-quantum key agreement and authentication protocols for satellite communications presents a cohesive narrative of technological evolution and security enhancement. Initially, Kumar and Garg [133] introduced an innovative scheme based on the learning with errors (LWE) problem tailored specifically for satellite networks to withstand quantum computer threats. Their protocol was designed to ensure user anonymity, defend against man-in-the-middle attacks, operate without time synchronization, and protect against smart card theft and password-guessing attacks. However, Dharminder et al. [134] identified vulnerabilities in this protocol, including weaknesses against smart card theft attacks, signal leakage attacks, and an improved version of the signal leakage attack. Following this, Mishra et al. [135] critiqued the work [134] for having too many communication rounds and being susceptible to denial-of-service attacks. In response, they developed a more efficient and quantum-safe authentication key exchange mechanism that utilizes timestamps to thwart replay attacks, optimizes the calculation process of authentication messages, reduces hash operations, and enables session key establishment within 4 message exchanges. Recently, Yadav et al. [136] pinpointed further weaknesses in the previous schemes, specifically key mismatch and offline dictionary attacks. They rectified these issues by generating and utilizing pseudo keys during the registration phase, thereby enhancing the protocol's robustness. Despite these improvements, Mishra and Pursharthi [137] found flaws in protocol [136] regarding session key consistency, computational inefficiency, and susceptibility to denial-of-service attacks. They proposed modifications such as altering hash computation methods, introducing timestamps, and minimizing redundant calculations, culminating in a streamlined and efficient authentication key exchange protocol that retained its security reduction to the LWE problem under the random oracle model.

Summary and lessons learned

In this section, we reviewed the security challenges and current state of satellite communication and space computing from both communication and data perspectives. From this analysis, we identified the following key insights:

- Space computing systems face critical security challenges including eavesdropping, signal jamming, and identity spoofing. Current cryptographic protocols suffer from high latency in centralized authentication and inadequate cross-domain security mechanisms. To address these limitations, decentralized authentication and lightweight key management solutions must be developed that maintain security while adapting to unique orbital constraints like dynamic network topologies and limited onboard resources.
- Blockchain technology enables decentralized identity management but requires computational overhead optimization. AE enhances data integrity but needs adaptation for space environments. Hybrid encryption schemes combining traditional algorithms with physical-layer security can meet the requirements of future space-terrestrial integrated networks. These technologies must be co-designed to effectively address the distinctive security challenges of space computing scenarios.

Challenges and Future Research Directions

Apart from the aforementioned issues, there are several other technical challenges in space computing that warrant attention.

These challenges are primarily reflected in various critical aspects, including the instability of communication links, which can disrupt data transmission rates and accuracy. Additionally, the complexity of processing and storing vast amounts of satellite and ground data necessitates efficient algorithms and substantial computational power. Furthermore, the heterogeneity of devices leads to interoperability issues, complicating collaboration. Real-time synchronization and resource constraints present ongoing obstacles that need innovative solutions for effective operation. Finally, security and privacy concerns, particularly regarding sensitive information, must be addressed.

- Instability of communication links: The communication links between satellites and ground stations are destabilized by many factors, such as the atmosphere, weather, and rotation of the Earth. This instability affects the rate and accuracy of data transmission and thus the speed and reliability of distributed collaborative computing.
- Heterogeneity and interoperability: Devices in the satellite distributed system may have different hardware and software configurations, leading to heterogeneity in data formats, communication protocols, and other aspects. How to realize interoperability between these heterogeneous devices and ensure that they can work together is an important technical challenge.
- Energy and computational resource constraints: Satellites devices are usually subject to energy and computational resource constraints, and how to realize efficient distributed collaborative computing under limited resource conditions is a problem that needs to be solved.
- Security and privacy protection: The air and space domain involves sensitive information such as national security and military secrets, so satellite-ground collaborative distributed reasoning must ensure data security and privacy protection. This includes technical challenges in data encryption, access control, and intrusion detection.

In order to effectively address the technical challenges facing space computing, it is crucial to undertake comprehensive research and development efforts focused on new algorithms, protocols, and system architectures. These advancements are essential for enhancing the performance and reliability of distributed collaborative computing systems, particularly in the context of satellite–satellite and satellite–Earth interactions.

System architectures

The design of innovative system architectures is crucial for space computing. These architectures must not only be flexible and scalable to accommodate different devices and technologies but also efficiently integrate collaborative work among ground-based large models, space lightweight large models, and small agent models. By optimizing the preprocessing capabilities of ground-based large models, the rapid response features of space lightweight large models, and the flexible decision-making of small model agents in specific scenarios, the entire system can achieve more efficient and precise processing of complex spatial computing tasks.

Stable communication

Adaptive modulation and coding (AMC) technology is crucial for optimizing data transmission in challenging environments like space. By dynamically adjusting modulation and coding schemes based on real-time link assessments, AMC enhances the stability and reliability of data transfer. This adaptability ensures efficient communication even under fluctuating conditions. Additionally, techniques like multipath transmission and redundant error correction further strengthen data transmission, creating a resilient communication framework essential for critical missions. These methods mitigate data loss and enable error correction without retransmission requests.

Processing of data, AI training, and inference in space

Adopting a distributed computing framework with parallel processing is essential for efficiently managing large volumes of aerospace big data. This approach facilitates simultaneous processing across multiple nodes, bringing data processing closer to the source. For instance, executing specific computational tasks directly on satellites considerably reduces data transmission latency and conserves valuable bandwidth resources. Furthermore, integrating distributed computing with ML algorithms enhances this process, enabling in-depth analysis and improved pattern recognition.

Addressing security and privacy

As space computing grows more intuitive and responsive, prioritizing security and privacy is essential. The sensitive data from space missions require stringent measures to protect against unauthorized access and cyber threats. Advanced encryption methods and secure communication protocols must be developed to safeguard this information. Additionally, privacy considerations should be integral to system design, ensuring compliance with international regulations. Establishing clear guidelines for data management will mitigate risks and enhance stakeholder confidence in space computing.

Future needs

In conclusion, to effectively tackle the technical challenges inherent in space computing, a multifaceted approach is required. This includes rigorous research and development of new algorithms, protocols, and system architectures, as well as fostering interdisciplinary collaboration and promoting innovation in related technologies.

Conclusion

This paper has highlighted major advancements in space computing, underscoring the critical roles of distributed storage systems, computational resource virtualization and scheduling, and AI techniques. We have identified key challenges, such as the instability of communication links, heterogeneity and interoperability, energy and resource constraints, and the need for security and privacy protection, which must be addressed to ensure the reliability and performance of space computing systems. Furthermore, we have explored emerging trends and future directions, including the integration of AI, edge computing, and distributed collaborative computing technologies, which promise to revolutionize the capabilities of space missions. As we move forward, it is essential for researchers and practitioners to collaborate across disciplines to develop innovative solutions that enhance the resilience and adaptability of

space computing systems. By doing so, we can unlock new possibilities for exploration, satellite operations, and scientific discovery in an increasingly complex space landscape.

Acknowledgments

We are grateful to F. Yu, L. Fei, T. Wang, S. Zhang, F. Xu, and M. Pang for their valuable help in gathering and organizing the materials used in the literature review.
Funding: This research was supported by Zhejiang Provincial Natural Science Foundation of China under grant no. LQ23F030009 and National Science Foundation for Young Scientists of China under grant no. 62302464.
Author contributions: All authors contributed equally to the collection, organization, and writing of the source material. E.X. and Z.Z. contributed equally to the overall writing and preparation of the manuscript.
Competing interests: The authors declare that they have no competing interests.

References

1. Liu J, Shi Y, Fadlullah ZM, Kato N. Space-air-ground integrated network: A survey. *IEEE Commun Surv Tutor*. 2018;20(4):2714–2741.
2. Xie R, Tang Q, Wang Q, Liu X, Yu FR, Huang T. Satellite-terrestrial integrated edge computing networks: Architecture, challenges, and open issues. *IEEE Netw*. 2020;34(3):224–231.
3. Shvachko K, Kuang H, Radia S, Chansler R. The Hadoop distributed file system. In: *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*. Piscataway (NJ): IEEE; 2010. p. 1–10.
4. Chang F, Dean J, Ghemawat S, Hsieh WC, Wallach DA, Burrows M, Chandra T, Fikes A, Gruber RE. Bigtable: A distributed storage system for structured data. *ACM Trans Comput Syst*. 2008;26(2):1–26.
5. Szuba M, Ameri P, Grabowski U, Meyer J, Streit A. A distributed system for storing and processing data from Earth-observing satellites: System design and performance evaluation of the visualisation tool. In: *2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*. Piscataway (NJ): IEEE; 2016. p. 169–174.
6. Huang H, Guo S, Wang K. Envisioned wireless big data storage for low-Earth-orbit satellite-based cloud. *IEEE Wirel Commun*. 2018;25(1):26–31.
7. Shanmugam K, Papailiopoulos DS, Dimakis AG, Caire G. A repair framework for scalar MDS codes. *IEEE J Sel Areas Commun*. 2014;32(5):998–1007.
8. Vajha M, Ramkumar V, Puranik B, Kini G, Lobo E, Sasidharan B, Kumar PV, Barg A, Ye M, Narayanamurthy S, Hussain S. Clay codes: Moulding MDS codes to yield an MSR code. In: *16th USENIX Conference on File and Storage Technologies (FAST 18)*. Oakland (CA): USENIX Association; 2018. p. 139–154.
9. Balaji S, Krishnan MN, Vajha M, Ramkumar V, Sasidharan B, Kumar PV. Erasure coding for distributed storage: An overview. *Sci China Inf Sci*. 2018;61(10):1–45.
10. Wang N, Wang Y, Gu S, Zhang Q, Xiang W. Adaptive data storage system for mobile satellite-terrestrial IoT. In: *2019 IEEE/CIC International Conference on Communications in China (ICCC)*. Oakland (CA): USENIX Association; IEEE; 2019. p. 106–111.
11. Wang F, Gu S, Zhang Q, Zhang N, Xiang W. Cost optimal regenerating codes design for satellite clustered distributed storage system. In: *2021 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE; 2021. p. 995–1000.
12. Pang B, Gu S, Zhang Q, Zhang N, Xiang W. CCOS: A coded computation offloading strategy for satellite-terrestrial integrated networks. In: *2021 International Wireless Communications and Mobile Computing (IWCMC)*. Piscataway (NJ): IEEE; 2021. p. 242–247.
13. Brinton CG, Aryafar E, Corda S, Russo S, Chiang M. An intelligent satellite multicast and caching overlay for CDNs to improve performance in video applications. In: *31st AIAA International Communications Satellite Systems Conference*. Washington (DC): American Institute of Aeronautics and Astronautics; 2013.
14. Kalantari A, Fittipaldi M, Chatzinotas S, Vu TX, Ottersten B. Cache-assisted hybrid satellite-terrestrial backhauling for 5G cellular networks. In: *GLOBECOM 2017—2017 IEEE Global Communications Conference*. Piscataway (NJ): IEEE; 2017. p. 1–6.
15. Vu TX, Maturo N, Vuppala S, Chatzinotas S, Grotz J. Efficient 5G edge caching over satellite. In: *36st AIAA International Communications Satellite Systems Conference*. Reston (VA): American Institute of Aeronautics and Astronautics; 2018.
16. Wu H, Li J, Lu H, Hong P. A two-layer caching model for content delivery services in satellite-terrestrial networks. In: *2016 IEEE Global Communications Conference (GLOBECOM)*. Piscataway (NJ): IEEE; 2016. p. 1–6.
17. Liu S, Hu X, Wang Y, Cui G, Wang W. Distributed caching based on matching game in LEO satellite constellation networks. *IEEE Commun Lett*. 2018;22(2):300–303.
18. Li J, Xue K, Wei DSL, Liu J, Zhang Y. Energy efficiency and traffic offloading optimization in integrated satellite/terrestrial radio access networks. *IEEE Trans Wirel Commun*. 2020;19(4):2367–2381.
19. Han D, Liao W, Peng H, Wu H, Wu W, Shen X. Joint cache placement and cooperative multicast beamforming in integrated satellite-terrestrial networks. *IEEE Trans Veh Technol*. 2021;71(3):3131–3143.
20. Zhang T, Wang Z, Liu Y, Xu W, Nallanathan A. Caching placement and resource allocation for cache-enabling UAV NOMA networks. In: *IEEE Transactions on Vehicular Technology*. Piscataway (NJ): IEEE; 2020.
21. Zhang T, Wang Z, Nallanathan XA. Joint resource, deployment, and caching optimization for AR applications in dynamic UAV NOMA networks. *IEEE Trans Wirel Commun*. 2022;21(5):3409–3422.
22. Li X, Zhang H, Zhou H, Wang N, Long K, Al-Rubaye S, Karagiannidis GK. Multi agent DRL for resource allocation and cache design in terrestrial-satellite networks. *IEEE Trans Wirel Commun*. 2023;22(8):5031–5042.
23. Dijkstra EW. A note on two problems in connexion with graphs. In: *Edsger Wybe Dijkstra: His Life,Work, and Legacy*. 1st ed. New York (NY): Association for Computing Machinery; 2022. p. 287–290.
24. Wood L, Clerget A, Andrikopoulos I, Pavlou G, Dabbous W. IP routing issues in satellite constellation networks. *Int J Satell Commun Netw*. 2001;19(1):69–92.
25. Hashimoto Y, Sarikaya B. Design of IP-based routing in a LEO satellite network. In: *Proceedings of the Third International Workshop on Satellite-Based Information*

Services (WORS-BIS'98). New York (NY): Association for Computing Machinery; 1998.
26. Werner M. A dynamic routing concept for ATM-based satellite personal communication networks. *IEEE J Sel Areas Commun*. 1997;15(8):1636–1648.
27. Chang HS, Kim BW, Lee CG, Min SL, Choi Y, Yang HS, Kim DN, Kim CS. FSA based link assignment and routing in low-Earth orbit satellite networks. *IEEE Trans Veh Technol*. 1997;47(3):1037–1048.
28. Gounder VV, Prakash R, Abu-Amara H. Routing in LEO-based satellite networks. In: *Wireless Communications and Systems, 2000. 1999 Emerging Technologies Symposium*. Piscataway (NJ): IEEE; 1999.
29. Fischer D, Basin D, Engel T. Topology dynamics and routing for predictable mobile networks. In: *2008 IEEE International Conference on Network Protocols*. Orlando (FL): IEEE; 2008.
30. Sigel E, Denby B, Le Hégarat-Mascle S. Application of ant colony optimization to adaptive routing in a LEO telecommunications satellite network. *Ann Télécommun*. 2002;**57**:520–539.
31. Liu S, Wu D, Zhang L. A routing model based on multiple-user requirements and the optimal solution. *IEEE Access*. 2020;8:156470–156483.
32. Zhao N, Long X, Wang J. A multi-constraint optimal routing algorithm in LEO satellite networks. *Wireless Netw*. 2021.
33. Tu Z, Zhou H, Li K, Li G, Shen Q. A routing optimization method for software-defined SGIN based on deep reinforcement learning. In: *2019 IEEE Globecom Workshops (GC Wk-shps)*. Piscataway (NJ): IEEE; 2019. p. 1–6.
34. Yin Y, Huang C, Wu DF, Huang S, Ashraf MWA, Guo Q. Reinforcement learning-based routing algorithm in satellite-terrestrial integrated networks. *Wirel Commun Mob Comput*. 2021;2021(1):3759631.
35. Na Z, Pan Z, Liu X, Deng Z, Gao Z, Guo Q. Distributed routing strategy based on machine learning for LEO satellite network. *Wirel Commun Mob Comput*. 2018;2018(1):3026405.
36. Liu D, Zhang J, Cui J, Ng SX, Maunder RG, Hanzo L. Deep learning aided routing for space-air-ground integrated networks relying on real satellite, flight, and shipping data. *IEEE Wirel Commun*. 2022;29(2):177–184.
37. Kato N, Fadlullah ZM, Tang F, Mao B, Tani S, Okamura A, Liu J. Optimizing space-airground integrated networks by artificial intelligence. *IEEE Wirel Commun*. 2019;26(4):140–147.
38. Wang F, Jiang D, Wang Z, Lv Z, Mumtaz S. Fuzzy-CNN based multi-task routing for integrated satellite-terrestrial networks. *IEEE Trans Veh Technol*. 2021;71(2):1913–1926.
39. Bogdan P, Dumitras T, Marculescu R. Stochastic communication: A new paradigm for fault-tolerant networks-on-chip. *VLSI Des*. 2007;2007(1):Article 095348.
40. Xue Y, Bogdan P. User cooperation network coding approach for NoC performance improvement. In: *Proceedings of the 9th International Symposium on Networks-on-Chip*. NOCS'15. Vancouver (BC, Canada): Association for Computing Machinery; 2015.
41. Siris VA, Ververidis CN, Polyzos GC, Liolis KP. Information-centric networking (ICN) architectures for integration of satellites into the future internet. In: *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*. Piscataway (NJ): IEEE; 2012. p. 1–6.
42. D'Oro S, Galluccio L, Morabito G, Palazzo S. SatCache: A profile-aware caching strategy for information-centric satellite networks. *Trans Emerg Telecommun Technol*. 2014;25(4):436–444.
43. An K, Li Y, Yan X, Liang T. On the performance of cache-enabled hybrid satellite terrestrial relay networks. *IEEE Wirel Commun Lett*. 2019;8(5):1506–1509.
44. Jiang C, Li Z. Decreasing big data application latency in satellite link by caching and peer selection. *IEEE Trans Netw Sci Eng*. 2020;7(4):2555–2565.
45. Qiu C, Yao H, Yu FR, Xu F, Zhao C. Deep Q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks. *IEEE Trans Veh Technol*. 2019;68(6):5871–5883.
46. Xu R, Di X, Chen J, Wang H, Luo H, Qi H, He X, Lei W, Zhang S. A hybrid caching strategy for information-centric satellite networks based on node classification and popular content awareness. *Comput Commun*. 2023;197:186–198.
47. Makris A, Kontopoulos I, Psomakelis E, Xyalis SN, Theodoropoulos T, Tserpes K. Towards a distributed storage framework for edge computing infrastructures. In: *Proceedings of the 2nd Workshop on Flexible Resource and Application Management on the Edge*. New York (NY): Association for Computing Machinery; 2022. p. 9–14.
48. Gheorghe AG, Crecana CC, Negru C, Pop F, and Dobre C. Decentralized storage system for edge computing. In: *2019 18th International Symposium on Parallel and Distributed Computing (ISPDC)*. Piscataway (NJ): IEEE; 2019. p. 41–49.
49. Ning H, Li Y, Shi F, Yang LT. Heterogeneous edge computing open platform and tools for internet of things. *Futur Gener Comput Syst*. 2020;106:67–76.
50. Gu J, Song S, Li Y, Luo H. GaiaGPU: Sharing GPUs in container clouds. In: *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*. Piscataway (NJ): IEEE; 2018. p. 469–476.
51. Liang G, Daud SN, Ismail NAB. Evolution of GPU virtualization to resource pooling. In: *Second International Conference on Electronic Information Technology (EIT 2023)*. Bellingham (WA): SPIE–The International Society for Optics and Photonics 2023. p. 641–650.
52. Jo HS, Lee MH, Choi DH. GPU virtualization using PCI direct pass-through. *Appl Mech Mater*. 2013;311:15–19.
53. Yang CT, Wang HY, Liu YT. Using PCI pass-through for GPU virtualization with CUDA. In: *IFIP International Conference on Network and Parallel Computing*. Berlin (Germany): Springer; 2012. p. 445–452.
54. Younge AJ, Walters JP, Crago S, Fox GC. Evaluating GPU passthrough in Xen for high performance cloud computing. In: *2014 IEEE International Parallel & Distributed Processing Symposium Workshops*. Piscataway (NJ): IEEE 2014. p. 852–859.
55. Hong CH, Spence I, Nikolopoulos DS. GPU virtualization and scheduling methods: A comprehensive survey. *ACM Comput Surv*. 2017;50(3):1–37.
56. Younge AJ, Walters JP, Crago SP, Fox GC. Supporting high performance molecular dynamics in virtualized clusters using IOMMU, SR-IOV, and GPUDirect. *ACM SIGPLAN Not*. 2015;50(7):31–38.
57. Yang X, Wang X, Yan L, Cao S. Technology for embedded GPU virtualization in the edge computing environment.

In: *2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*. Piscataway (NJ): IEEE; 2022. p. 1657–1663.
58. Reaño C, Silla F, Duato J. Enhancing the rCUDA remote GPU virtualization frame work: From a prototype to a production solution. In: *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. Piscataway (NJ): IEEE; 2017. p. 695–698.
59. Silla F, Prades J, Baydal E, Reaño C. Improving the performance of physics applications in atom-based clusters with rCUDA. *J Parallel Distrib Comput*. 2020;137:160–178.
60. Chien S, Rabideau G, Knight R, Sherwood R, Engelhardt B, Mutz D, Estlin T, Smith B, Fisher F, Barrett T, et al. ASPEN—Automated planning and scheduling for space mission operations. In: *SpaceOps 2000*. Paris (France): CNES; 2000.
61. Cesta A, Fratini S, Donati A, Oliveira H, Policella N. Rapid prototyping of planning scheduling tools. In: *2009 Third IEEE International Conference on Space Mission Challenges for Information Technology*. Piscataway (NJ): IEEE; 2009. p. 270–277.
62. Carrel AR, Palmer PL. An evolutionary algorithm for near-optimal autonomous resource management. In: *8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*. Palo Alto (CA): 2005. p. 25.
63. Lu J, Chen Y, He R. A learning-based approach for agile satellite onboard scheduling. *IEEE Access*. 2020;8:16941–16952.
64. Zhang Y, Chen C, Liu L, Lan D, Jiang H, Wan S. Aerial edge computing on orbit: A task offloading and allocation scheme. *IEEE Trans Netw Sci Eng*. 2022;10(1):275–285.
65. Zeleke DA, Kim HD. A new strategy of satellite autonomy with machine learning for efficient resource utilization of a standard performance CubeSat. *Aerospace*. 2023;10(1):78.
66. Wang F, Jiang D, Qi S, Qiao C, Shi L. A dynamic resource scheduling scheme in edge computing satellite networks. *Mob Netw Appl*. 2021;26(2):597–608.
67. Dang VD, Dash RK, Rogers A, Jennings NR. Overlapping coalition formation for efficient data fusion in multi-sensor networks. In: *Proceedings of the National Conference on Artificial Intelligence, AIPS. 2006*. Palo Alto (CA): AAAI Press; 2006. p. 635–640.
68. Goradia HJ, Vidal JM. An equal excess negotiation algorithm for coalition formation. In: *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*. AAMAS '07. Honolulu (HI): ACM; 2007.
69. Pralet C, Verfaillie G. Using constraint networks on timelines to model and solve planning and scheduling problems. In: *Proceedings of the 18th International Conference on Artificial Intelligence Planning and Scheduling (ICAPS 2008)*. Palo Alto (CA): AAAI Press; 2008.
70. Abdallah S, Lesser V. Organization-based cooperative coalition formation. In: *Proceedings IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2004. (IAT 2004)*. Piscataway (NJ): IEEE; 2004. p. 162–168.
71. Sims M, Goldman CV, Lesser V. Self-organization through bottom-up coalition formation. In: *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*. AAMAS '03. Melbourne (Australia): Association for Computing Machinery; 2003. p. 867–874.
72. Iacopino C, Palmer P, Policella N, Donati A, Brewer A. How ants can manage your satellites. *Acta Futura*. 2014;9:57–70.
73. Tang Q, Xie R, Fang Z, Huang T, Chen T, Zhang R, Yu FR. Joint service deployment and task scheduling for satellite edge computing: A two-timescale hierarchical approach. *IEEE J Sel Areas Commun*. 2024;42(5):1063–1079.
74. Han J, Wang H, Wu S, Wei J, Yan L. Task scheduling of high dynamic edge cluster in satellite edge computing. In: *2020 IEEE World Congress on Services (SERVICES)*. Piscataway (NJ): IEEE; 2020. p. 287–293.
75. Yang W, He L, Liu X, Chen Y. Onboard coordination and scheduling of multiple autonomous satellites in an uncertain environment. *Adv Space Res*. 2021;68(11):4505–4524.
76. Richards RA, Houlette RT, Mohammed JL. Distributed satellite constellation planning and scheduling. In: *FLAIRS*. Palo Alto (CA): AAAI Press; 2001. p. 68–72.
77. Chong W, Jun L, Ning J, Jun W, Hao C. A distributed cooperative dynamic task planning algorithm for multiple satellites based on multi-agent hybrid learning. *Chin J Aeronaut*. 2011;24(4):493–505.
78. Yao F, Li J, Chen Y, Chu X, Zhao B. Task allocation strategies for cooperative task planning of multi-autonomous satellite constellation. *Adv Space Res*. 2019;63(2):1073–1084.
79. Bonnet J, Gleizes MP, Kaddoum E, Rainjonneau S, Flandin G. Multi-satellite mission planning using a self-adaptive multi-agent system. In: *2015 IEEE 9th International Conference on Self-adaptive and Self-organizing Systems*. Piscataway (NJ): IEEE; 2015. p. 11–20.
80. Tang Q, Fei Z, Li B, Han Z. Computation offloading in LEO satellite networks with hybrid cloud and edge computing. *IEEE Internet Things J*. 2021;8(11):9164–9176.
81. Li H, Chen C, Huang C, Cong L, Gao Z. Collaborative resource orchestration for on orbit edge computing in 5G SAGIN. In: *Proceedings of the 1st ACM MobiCom Workshop on Satellite Networking and Computing*. New York (NY): ACM; 2023. p. 13–18.
82. He J, Cheng N, Yin Z, Zhou H, Zhou C, Aldubaikhy K, Alqasir A, Shen XS. Load-aware network resource orchestration in LEO satellite network: A GAT-based approach. *IEEE Internet Things J*. 2024;11(9):15969–15984.
83. Chai F, Zhang Q, Yao H, Xin X, Gao R, Guizani M. Joint multi-task offloading and resource allocation for mobile edge computing systems in satellite IoT. *IEEE Trans Veh Technol*. 2023;72(6):7783–7795.
84. Zhang S, Cui G, Long Y, Wang W. Joint computing and communication resource allocation for satellite communication networks with edge computing. *China Commun*. 2021;18(7):236–252.
85. Qin X, Ma T, Tang Z, Zhang X, Zhou H, Zhao L. Service-aware resource orchestration in ultra-dense LEO satellite-terrestrial integrated 6G: A service function chain approach. *IEEE Trans Wirel Commun*. 2023;22(9):6003–6017.
86. Ding C, Wang JB, Zhang H, Lin M, Li GY. Joint optimization of transmission and computation resources for satellite and high altitude platform assisted edge computing. *IEEE Trans Wirel Commun*. 2022;21(2):1362–1377.
87. Li X, Yang J, Fan H. Dynamic network resource autonomy management and task scheduling method. *Mathematics*. 2023;11(5):1232.
88. He J, Cheng N, Yin Z, Zhou C, Zhou H, Quan W, Lin XH. Service-oriented network resource orchestration in space-air-ground integrated network. *IEEE Trans Veh Technol*. 2023;73(1):1162–1174.

89. Giuffrida G, Fanucci L, Meoni G, Batič M, Buckley L, Dunne A, Van Dijk C, Esposito M, Hefele J, Vercruyssen N, et al. The Φ-Sat-1 mission: The first on-board deep neural network demonstrator for satellite Earth observation. *IEEE Trans Geosci Remote Sens*. 2021;60:1–14.
90. McMahan B, Moore E, Ramage D, Hampson S, Arcas BA y. Communication-efficient learning of deep networks from decentralized data. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Cambridge (MA): JMLR Inc. (Proceedings of Machine Learning Research); 2017. p. 1273–1282.
91. Zhai Z, Wu Q, Yu S, Li R, Zhang F, Chen X. FedLEO: An offloading-assisted decentralized federated learning framework for low Earth orbit satellite networks. *IEEE Trans Mob Comput*. 2023;23(5):5260–5279.
92. Zhang H, Zhao H, Liu R, Gao X, Xu S. Leader federated learning optimization using deep reinforcement learning for distributed satellite edge intelligence. *IEEE Trans Serv Comput*. 2024;17(5):2544–2557.
93. Han DJ, Hosseinalipour S, Love DJ, Chiang M, Brinton CG. Cooperative federated learning over ground-to-satellite integrated networks: Joint local computation and data offloading. *IEEE J Sel Areas Commun*. 2024;42(5):1080–1096.
94. Elmahallawy M, Luo T, Ramadan K. Communication-efficient federated learning for LEO satellite networks integrated with HAPs using hybrid NOMA-OFDM. *IEEE J Sel Areas Commun*. 2024;42:1097–1114.
95. Lin Z, Zhu G, Deng Y, Chen X, Gao Y, Huang K, Fang Y. Efficient parallel split learning over resource-constrained wireless edge networks. *IEEE Trans Mob Comput*. 2024;32(10):9224–9239.
96. Jiang W, Han H, Zhang Y, Mu J. Federated split learning for sequential data in satellite–terrestrial integrated networks. *Inf Fusion*. 2024;103:Article 102141.
97. Lin Z, Qu G, Chen X, Huang K. Split learning in 6G edge networks. *IEEE Wirel Commun*. 2024;31(4):170–176.
98. Peng S, Hou X, Shen Z, Zheng Q, Jin J, Tagami A, Yuan J. Collaborative satellite computing through adaptive DNN task splitting and offloading. arXiv. 2024. https://doi.org/10.48550/arXiv.2405.03181
99. Guan J, Zhang Q, Murturi I, Donta PK, Dustdar S, Wang S. Collaborative inference in DNN-based satellite systems with dynamic task streams. In: *ICC 2024-IEEE International Conference on Communications*. Piscataway (NJ): IEEE; 2024. p. 3803–3808.
100. Sun J, Wu C, Mumtaz S, Tao J, Cao M, Wang M, Frascolla V. An efficient privacy-aware split learning framework for satellite communications. *IEEE J Sel Areas Commun*. 2024;42(12):3355–3365.
101. Lin Z, Chen Z, Fang Z, Chen X, Wang X, Gao Y. Fedsn: A federated learning framework over heterogeneous LEO satellite networks. *IEEE Trans Mob Comput*. 2024;24(3):1293–1307.
102. Wang G, Zhang N, Wang J, Liu W, Xie Y, Chen H. Knowledge distillation-based lightweight change detection in high-resolution remote sensing imagery for on-board processing. *IEEE J Sel Top Appl Earth Obs Remote Sens*. 2024;17:3860–3877.
103. Pang Y, Zhang Y, Wang Y, Wei X, Chen B. Exploring model compression limits and laws: A pyramid knowledge distillation framework for satellite-on-orbit object recognition. *IEEE Trans Geosci Remote Sens*. 2024;62:5603313.
104. Elmahallawy M, Luo T. One-shot federated learning for LEO constellations that reduces convergence time from days to 90 minutes. In: *2023 24th IEEE International Conference on Mobile Data Management (MDM)*. Piscataway (NJ): IEEE; 2023. p. 45–54.
105. Ingemarsson I, Wong C. Encryption and authentication in on-board processing satellite communication systems. *IEEE Trans Commun*. 1981;29(11):1684–1687.
106. Cruickshank H. A security system for satellite networks. In: *Fifth International Conference on Satellite Systems for Mobile Communications and Navigation*. London (UK): IET; 1996. p. 187–190.
107. Ortakci Y, Abdullah MY. Performance analyses of AES and 3DES algorithms for encryption of satellite images. In: Ben Ahmed M, Rakıp Karas, I, Santos D, Sergeyeva O, Boudhir AA, editors. *Innovations in Smart Cities Applications Volume 4*. Cham: Springer International Publishing; 2021. p. 877–890.
108. Ning L, Kanfeng L, Wenliang L, Zhongliang D. A joint encryption and error correction method used in satellite communications. *China Commun*. 2014;11(3):70–79.
109. Murtaza A, Pirzada SJH, Hasan MN, Xu T, Jianwei L. An efficient encryption algorithm for perfect forward secrecy in satellite communication. In: *Advances in Cyber Security: First International Conference, ACeS 2019, Penang, Malaysia, July 30–August 1, 2019, Revised Selected Papers 1*. Berlin (Germany): Springer; 2020. p. 289–302.
110. Jeon S, Kwak J, Choi JP. Advanced multibeam satellite network security with encryption and beamforming technologies. In: *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*. Piscataway (NJ): IEEE; 2022. p. 1177–1182.
111. Hussain Pirzada SJ, Murtaza A, Jianwei L, Xu T. The parallel CMAC authenticated encryption algorithm for satellite communication. In: *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*. Piscataway (NJ): IEEE; 2019. p. 1–5.
112. Tawfik AA, Elbayoumy AD, Hussein M, Elghandour AH. A new security mechanism for MIL-STD-1553 using authenticated encryption algorithms. In: *2024 6th International Conference on Computing and Informatics (ICCI)*. Piscataway (NJ): IEEE; 2024. p. 115–119.
113. Yue P, An J, Zhang J, Pan G, Wang S, Xiao P, Hanzo L. On the security of LEO satellite communication systems: Vulnerabilities, countermeasures, and future trends. TechRxiv. 2022. https://www.doi.org/10.36227/techrxiv.18093941.v1
114. Wang C, Zhang Y, Zhang Q, Xu X, Chen W, Li H. SE-CAS: Secure and efficient cross domain authentication scheme based on blockchain for space TT&C networks. *IEEE Internet Things J*. 2024;11(16):26806–26818.
115. Xiong T, Zhang R, Liu J, Huang T, Liu Y, Yu FR. A blockchain-based and privacy preserved authentication scheme for inter-constellation collaboration in space-ground integrated networks. *Comput Netw*. 2022;206:Article 108793.
116. Guo J, Chang L, Song Y, Yao S, Zheng Z, Hao Y, Zhu S, Guo W, Zhao M. AHA-BV: Access and handover authentication protocol with batch verification for satellite–terrestrial integrated networks. *Comput Stand Interfaces*. 2025;91:Article 103870.
117. Xue K, Meng W, Zhou H, Wei DSL, Guizani M. A lightweight and secure group key based handover authentication protocol

for the software-defined space information network. *IEEE Trans Wirel Commun*. 2020;19(6):3673–3684.
118. Lai C, Chen Z. Group-based handover authentication for space-air-ground integrated vehicular networks. In: *ICC 2021—IEEE International Conference on Communications*. Piscataway (NJ): IEEE; 2021. p. 1–6.
119. Yang Y, Cao J, Ma R, Cheng L, Chen L, Niu B, Li H. FHAP: Fast handover authentication protocol for high-speed mobile terminals in 5G satellite–terrestrial-integrated networks. *IEEE Internet Things J*. 2023;10(15):13959–13973.
120. Li S, Liu M, Wei S. A distributed authentication protocol using identity-based encryption and blockchain for LEO network. In: Wang G, Atiquzzaman M, Yan Z, Choo KKR, editors. *Security, Privacy, and Anonymity in Computation, Communication, and Storage*. Cham: Springer International Publishing; 2017. p. 446–460.
121. Guan J, Wu Y, Yao S, Zhang T, Su X, Li C. BSLA: Blockchain-assisted secure and lightweight authentication for SGIN. *Comput Commun*. 2021;176:46–55.
122. Ma R, Cao J, Feng D, Li H. LAA: Lattice-based access authentication scheme for IoT in space information networks. *IEEE Internet Things J*. 2020;7(4):2791–2805.
123. Yang Q, Xue K, Xu J, Wang J, Li F, Yu N. AnFRA: Anonymous and fast roaming authentication for space information network. *IEEE Trans Inf Forensics Secur*. 2019;14(2):486–497.
124. Liu X, Yang A, Huang C, Li Y, Li T, Li M. Decentralized anonymous authentication with fair billing for space-ground integrated networks. *IEEE Trans Veh Technol*. 2021;70(8):7764–7777.
125. Liu Y, Ni L, Peng M. A secure and efficient authentication protocol for satellite terrestrial networks. *IEEE Internet Things J*. 2023;10(7):5810–5822.
126. Ntanos A, Zavitsanos D, Giannoulis G, Avramopoulos H. Satellite assisted QKD key encapsulation. In: *ICC 2022—IEEE International Conference on Communications*. Piscataway (NJ): IEEE; 2022. p. 3251–3256.
127. Vergoossen T, Loarte S, Bedington R, Kuiper H, Ling A. Modelling of satellite constellations for trusted node QKD networks. *Acta Astronaut*. 2020;173:164–171.
128. Huang D, Zhao Y, Yang T, Rahman S, Yu X, He X, Zhang J. Quantum key distribution over double-layer quantum satellite networks. *IEEE Access*. 2020;8:16087–16098.
129. Yantao Z, Jianfeng M. A highly secure identity-based authenticated key-exchange protocol for satellite communication. *J Commun Netw*. 2010;12(6):592–599.
130. Song IA, Lee YS. Improvement of key exchange protocol to prevent man-in-the-middle attack in the satellite environment. In: *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*. Piscataway (NJ): IEEE Press; 2016. p. 408–413.
131. Altaf I, Saleem MA, Mahmood K, Kumari S, Chaudhary P, Chen CM. A lightweight key agreement and authentication scheme for satellite-communication systems. *IEEE Access*. 2020;8:46278–46287.
132. Murtaza A, Xu T, Pirzada SJH, Liu J. A lightweight authentication and key sharing protocol for satellite communication. *Int J Comput Commun Control*. 2020;9:46–53.
133. Kumar U, Garg M. Learning with error-based key agreement and authentication scheme for satellite communication. *Int J Satell Commun Netw*. 2022;40(2):83–95.
134. Dharminder D, Dadsena PK, Gupta P, Sankaran S. A post-quantum secure construction of an authentication protocol for satellite communication. *Int J Satell Commun Netw*. 2023;41(1):14–28.
135. Mishra D, Rewal P, Pursharthi K. Efficient and quantum-secure authenticated key exchange scheme for mobile satellite communication networks. *Int J Satell Commun Netw*. 2024;42(4):313–328.
136. Yadav S, Dabra V, Malik P, Kumari S. Flaw and amendment of Dharminder et al.'s authentication protocol for satellite communication. *Security Privacy*. 2024;7(4):Article e383.
137. Mishra D, Pursharthi K. Cryptanalysis with improvement on lattice-based authenticated key exchange protocol for mobile satellite communication networks. *Security Privacy*. 2024;7(5):Article e407.
138. Xiao Y, Ye Z, Wu M, Li H, Xiao M, Alouini MS, Al-Hourani A, Cioni S. Space-air ground integrated wireless networks for 6G: Basics, key technologies, and future trends. *IEEE J Sel Areas Commun*. 2024;42(12):3327–3354.
139. Guo H, Li J, Liu J, Tian N, Kato N. A survey on space-air-ground-sea integrated network security in 6G. *IEEE Commun Surv Tutor*. 2022;24(1):53–87.
140. Sharif S, Zeadally S, Ejaz W. Space-aerial-ground-sea integrated networks: Resource optimization and challenges in 6G. *J Netw Comput Appl*. 2023;215:Article 103647.
141. Cui H, Zhang J, Geng Y, Xiao Z, Sun T, Zhang N, Liu J, Wu Q, Cao X. Space-air ground integrated network (SAGIN) for 6G: Requirements, architecture and challenges. *China Commun*. 2022;19(2):90–108.
142. Cheng N, Jingchao HE, Zhisheng YI, Conghao ZH, Huaqing WU, Feng LY, Haibo ZH, Xuemin SH. 6G service-oriented space-air-ground integrated network: A survey. *Chin J Aeronaut*. 2022;35(9):1–18.
143. Ray PP. A review on 6G for space-air-ground integrated network: Key enablers, open challenges, and future direction. *J King Saud Univ Comput Inf Sci*. 2022;34(9):6949–6976.
144. Azari MM, Solanki S, Chatzinotas S, Kodheli O, Sallouha H, Colpaert A, Montoya JF, Pollin S, Haqiqatnejad A, Mostaani A, et al. Evolution of non-terrestrial networks from 5G to 6G: A survey. *IEEE Commun Surv Tutor*. 2022;24(4):2633–2672.
145. Shen Z, Jin J, Tan C, Tagami A, Wang S, Li Q, Zheng Q, Yuan J. A survey of next generation computing technologies in space-air-ground integrated networks. *ACM Comput Surv*. 2023;56(1):1–40.
146. Speretta S, Ilin A. Scalable data processing system for satellite data mining. In: *68th International Astronautical Congress: Unlocking Imagination, Fostering Innovation and Strengthening Security*. Paris (France): International Astronautical Federation;; 2017.
147. Absardi ZN, Javidan R. Classification of big satellite images using Hadoop clusters for land cover recognition. In: *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*. Piscataway (NJ): IEEE; 2017. p. 600–603.
148. Chen G, Nie P, Jing W. A distributed storage scheme for remote sensing image based on Mapfile. *Int J Perform Eng*. 2018;14:Article 2545.
149. Wang H, Tang X, Shi S, Ye F. Research on the construction of data management system of massive satellite images. In: *2018 International Workshop on Big Geospatial Data and Data Science (BGDDS)*. Piscataway (NJ): IEEE; 2018. p. 1–4.
150. Brown PG. Overview of SciDB: Large scale array storage, processing and analysis. In: *Proceedings of the 2010 ACM*

SIGMOD International Conference on Management of Data. New York (NY): Association for Computing Machinery; 2010. p. 963–968.
151. Krčál L, Ho SS. A SciDB-based framework for efficient satellite data storage and query based on dynamic atmospheric event trajectory. In: *Proceedings of the 4th International ACM SIGSPATIAL Workshop on Analytics for Big Geospatial Data*. New York (NY): Association for Computing Machinery; 2015. p. 7–14.
152. Xu R, Gu D, Zhang J, Zhu D, Lin M. Research on satellite-borne big-data storage system. In: *Proceedings of the 2022 4th International Conference on Big-data Service and Intelligent Computation*. New York (NY): Association for Computing Machinery; 2022. p. 36–42.