INTERNATIONAL JOURNAL OF RESEARCH AND INNOVATION IN APPLIED SCIENCE (IJRIAS)
ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VI June 2026

Topology-Aware Learning for Routing In Satellite–Terrestrial Integrated Networks: A Review of Graph Neural Network and Reinforcement Learning Approaches

Eyeneka J. Ntuen, Akaninyene B. Obot, Kufre M. Udofia*, Unwana I. Ibanga

Department of Electrical and Electronic Engineering, University of Uyo Nigeria

DOI: https://doi.org/10.51584/IJRIAS.2026.11060214

Received: 14 June 2026; Accepted: 19 June 2026; Published: 10 July 2026

ABSTRACT

Satellite–Terrestrial Integrated Networks (STINs) are a key part of 5G-Advanced and new 6G non-terrestrial networks. They connect the world by combining LEO, MEO, and GEO satellite constellations with ground-based infrastructure. But routing is very hard because of highly dynamic topologies, different link characteristics, and large-scale networks. This makes traditional protocols and topology-agnostic learning methods less useful. This review analyses topology-aware learning-based routing for STINs, concentrating on Graph Neural Networks (GNNs) and hybrid GNN–Reinforcement Learning (GNN–RL) frameworks. By modelling STINs as graphs that change over time, these methods clearly show how relationships and multi-hop interactions work, which are important for routing that can grow and change. Comprehensive analyses are conducted on classical routing, non-topology-aware reinforcement learning, purely GNN-based methodologies, and hybrid GNN–RL architectures, emphasising their merits and drawbacks in dynamic satellite–terrestrial contexts. We also look at hierarchical and multi-agent extensions, as well as current datasets and evaluation methods. Finally, important open problems related to scalability, non-stationarity, and real-world use are found, and future research directions that fit with new 6G non-terrestrial network standards are laid out.

Keywords: 5G-advanced, 6G non-terrestrial networks, Satellite–Terrestrial Integrated Networks, GNN, RL

INTRODUCTION

Satellite–Terrestrial Integrated Networks (STINs) have become an important part of the architecture of 5G-Advanced and 6G non-terrestrial networks (NTNs). They make it possible for satellite constellations to connect with terrestrial access and core networks, allowing for seamless global connectivity. STINs want to support widespread broadband access, ultra-reliable low-latency communications, and huge connectivity for underserved and remote areas by combining Low Earth Orbit (LEO), Medium Earth Orbit (MEO), and Geostationary Earth Orbit (GEO) satellites with ground infrastructure (Kodheli et al., 2021). However, combining space and ground segments creates new problems at the network layer, especially for routing, because the topologies are always changing, the link characteristics are different, and the network is very large (Liu et al., 2023).

STINs are different from regular terrestrial networks in that their topologies change all the time, but quickly. This is because of satellite orbital motion, the establishment of dynamic intersatellite links (ISLs), and the fact that satellite–ground visibility changes over time. These traits present significant challenges to traditional routing mechanisms that depend on relatively unchanging graph assumptions and regular link-state updates. In large LEO constellations, routing tables can become out of date quickly due to frequent changes in the topology. This can cause more signalling overhead, less optimal path selection, and route instability (Chen et al., 2025). Consequently, routing in STINs must adjust to both traffic fluctuations and ongoing structural modifications in network connectivity.

To tackle these challenges, learning-based routing methodologies, especially reinforcement learning (RL) and deep reinforcement learning (DRL), have garnered increasing attention for satellite and integrated networks.

Reinforcement learning (RL) methods let agents learn routing policies by having them interact with the environment. This means that they can change as the network changes without needing clear analytical models (Zhu et al., 2021). Despite this, a lot of current RL-based routing schemes use state representations that don't depend on topology and use flat feature vectors like queue lengths, link delays, or local traffic statistics. These representations work well in small or static situations, but they don't show the relational and multi-hop dependencies that are common in STINs. This makes them hard to scale, slow to converge, and not very useful for networks of different sizes or configurations (Roth et al., 2024).

As a result, being aware of topology has become a key need for smart routing in STINs. From a graph-theoretic standpoint, STINs can be effectively represented as time-evolving graphs, with nodes symbolising satellites, gateways, or terrestrial routers, and edges denoting communication links that change dynamically and possess diverse attributes. By explicitly including this graph structure in routing decisions, algorithms can think about things like neighbourhood relationships, path diversity, and global connectivity patterns. These are things that flat-state learning methods don't usually do (Wu et al., 2021). This understanding has led to a greater interest in Graph Neural Networks (GNNs), which are made to work with graph-structured data and learn topology-aware representations through message passing and neighbourhood aggregation.

Recent research has shown that GNN-based routing frameworks and hybrid architectures that combine GNNs with RL greatly improve routing performance in dynamic satellite and integrated networks by making them more general, robust, and scalable (Xu et al., 2024; Hu et al., 2024). In these kinds of hybrid frameworks, GNNs are often used to turn the changing network topology into small embeddings that RL agents can then use to improve long-term routing policies. This combination is especially appealing for STINs, where routing choices have to take into account both structural dependencies and making decisions in a sequence when there is uncertainty.

Even with these improvements, the current literature is still disjointed, with studies that use different system assumptions, topology models, learning architectures, and evaluation methods. Furthermore, the majority of existing surveys examine STINs from overarching architectural or standardisation perspectives, providing insufficient analysis of the relative advantages and disadvantages of topology-aware learning-based routing methodologies. This review fills this gap by looking closely and systematically at routing intelligence in STINs, with a focus on topology-aware methods that use GNNs and hybrid GNN–RL frameworks. This review seeks to elucidate contemporary research trends, pinpoint unresolved challenges, and delineate promising avenues for future investigations into intelligent routing within next-generation satellite-terrestrial networks by conceptualising STIN routing as a dynamic graph learning problem. Fig. 1 shows how Satellite–Terrestrial Integrated Networks are set up as a whole.

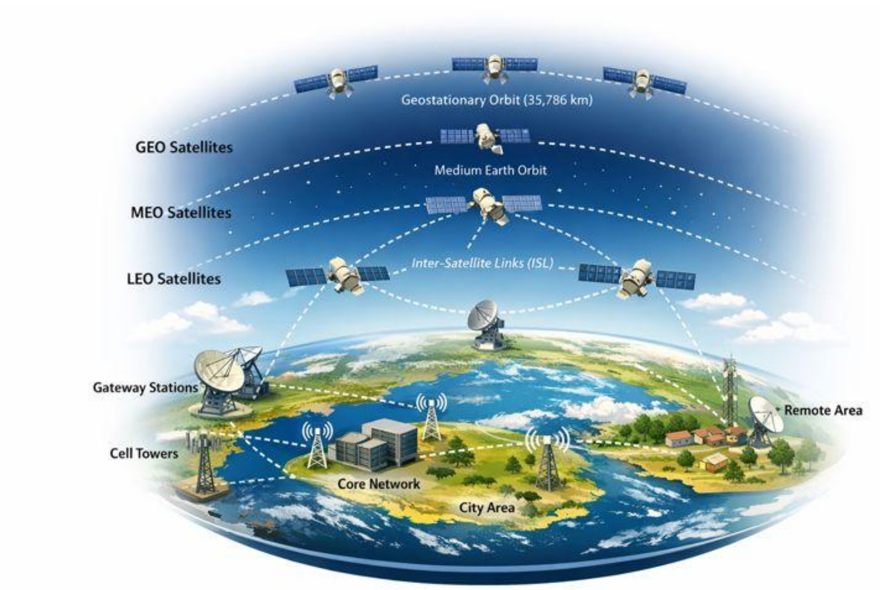

Fig. 1. Overall Architecture of a STIN

Satellite–Terrestrial Integrated Networks As Dynamic Graph Systems

Architectural Overview of Satellite–Terrestrial Integrated Networks

The space and ground segments of STINs are tightly linked so that they can provide seamless end-to-end connectivity. The space segment usually has LEO, MEO, and GEO satellite constellations that are connected by inter-satellite links (ISLs). The ground segment has gateway stations, terrestrial backhaul, and access networks (Giordani et al., 2020). LEO satellites have low latency and high spatial reuse, but they also change the topology quickly. GEO satellites, on the other hand, cover a wide area and have relatively stable connections, but they take longer to send and receive signals. Kodheli et al. (2021) said that routing in STINs must work across different layers with different mobility patterns, latency profiles, and control timescales.

Graph Modelling of STINs

From a network-theoretic point of view, STINs can be thought of as time-varying graphs, with nodes standing for satellites, gateways, or terrestrial routers and edges standing for communication links whose availability and properties change over time (Ekici et al., 2022). Edge weights usually represent delay, capacity, or reliability, while node features may represent traffic load or energy limits. Because satellites can move around, the connectivity graph changes all the time, which means that either snapshot-based or time-expanded graph representations are needed. This graph abstraction gives us a single way to represent the spatial relationships, multi-hop dependencies, and cross-layer interactions that are common in STIN routing problems (Wu et al., 2021). Figure 2 shows this even more clearly.

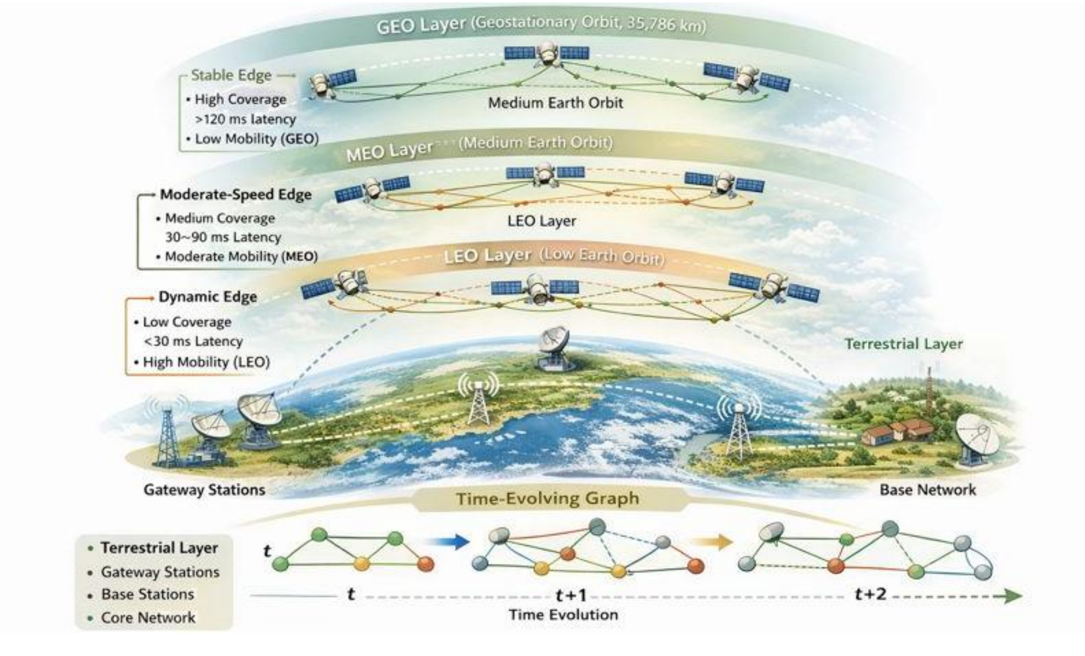

Fig. 2. STIN as a Time-Evolving Graph Model

Routing Challenges Arising from Dynamic Graphs

As shown in Fig. 3, the fact that STINs can change makes routing more difficult in a number of ways. Frequent changes in topology cause links to be available only sometimes and paths to be possible only at certain times, which makes static shortest-path computation useless (Zhang et al., 2023). Large constellation sizes make signalling overhead and state-space complexity worse, and partial observability makes it harder for individual nodes to get global topology information (Liu et al., 2023). These factors together make it hard for traditional routing protocols to scale, stay stable, and adapt. This is why learning-based and topology-aware routing methods that can reason over changing graph structures are becoming more popular.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VI June 2026

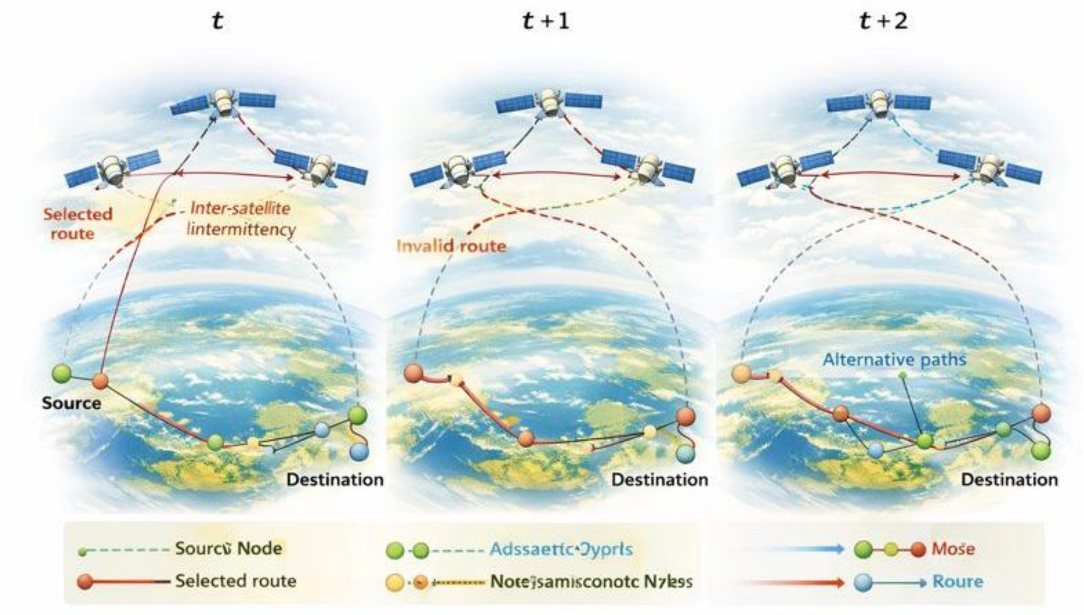

Fig. 3. Routing Challenges Induced by Dynamic STIN Topologies

Classical And Non–Topology-Aware Learning-Based Routing

Classical Routing in Satellite and STIN Environments

Classical routing methods for satellite and satellite-terrestrial integrated networks are mostly based on terrestrial link-state and distance-vector protocols, which have been changed to work with satellite mobility and long propagation delays (Ekici et al., 2022). Many people have used shortest-path algorithms like Dijkstra and its variations with either periodically updated topology information or precomputed routing tables based on deterministic orbital models. These methods can take advantage of the fact that satellites move in a predictable way, but they require a lot of extra signalling and have trouble keeping routes optimal in large LEO constellations where link availability and traffic conditions change quickly (Kodheli et al., 2021). In addition, classical routing usually sees the network as a static or almost static graph, which makes it less able to respond to temporary congestion and failures.

Reinforcement Learning–Based Routing Approaches

Reinforcement learning (RL) has been investigated as an alternative to traditional routing because of its capacity to modify routing decisions through engagement with a dynamic environment. In RL-based routing, agents learn rules that connect the network states they see to forwarding actions. The goal is to improve long-term performance metrics like latency, throughput, or packet delivery ratio. Deep Q-Networks (DQNs) and policy-gradient methods are two types of DRL techniques that have been used in both satellite-only and satellite–terrestrial situations. These techniques are more adaptable than static routing schemes (Zhu et al., 2021; Zhang et al., 2023). These methods lessen the need for explicit topology dissemination and can better adapt to changing traffic patterns than traditional methods.

Limitations of Non–Topology-Aware Learning Methods

Even though they can be used in many different ways, many RL-based routing methods use topology-agnostic state representations that depend on local metrics like queue lengths, link delays, and neighbour statistics. These flat representations do not show the relational structure and multi-hop dependencies that are common in STINs. This makes them less scalable and less generalisable as the network size grows, as Liu et al. (2023) point out. Large constellations often have unstable training and slow convergence because the state space grows too quickly and the environments change too quickly. Additionally, policies acquired in one network configuration frequently fail to transfer efficiently to alternative constellation arrangements or traffic conditions, underscoring the necessity for explicit topology-aware learning mechanisms (Roth et al., 2024). Figure 4 shows the comparison.

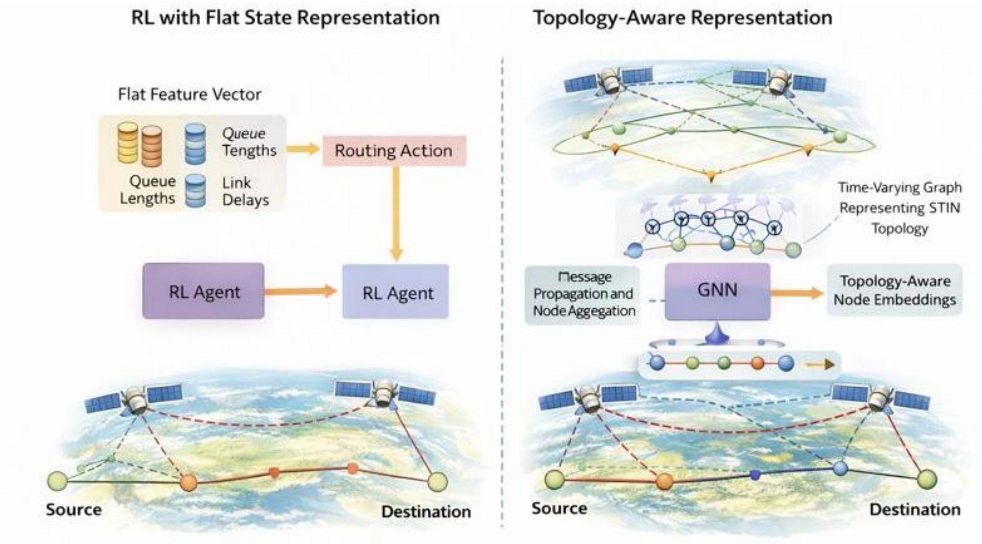

Fig. 4. Flat State Representation vs. Topology-Aware Representation

Graph Neural Networks for Topology-Aware Networking

Fundamentals of Graph Neural Networks

GNNs are a type of deep learning model that works directly on graph-structured data by using information about how nodes are connected and how they relate to each other. Most GNN architectures use a message-passing model, where each node learns topology-aware representations by repeatedly gathering data from its neighbours. GNNs use stacked aggregation layers to get both local and global structural properties of the graph. This lets them generalise across graphs of different sizes and shapes (Wu et al., 2021). These traits make GNNs great for communication networks, where performance depends on interactions between multiple hops instead of just link metrics. Figure 5 shows how messages are passed in GNN.

GNNs in Communication Networks

GNNs are being used more and more to solve networking problems like routing, controlling congestion, predicting traffic, and allocating resources. GNN-based models can learn policies that adapt to changes in structure without needing hand-crafted features by encoding network topology and link attributes into node or edge embeddings (Scarselli et al., 2009). GNNs have been demonstrated to enhance scalability and robustness in dynamic network environments, particularly when network size or connectivity fluctuates over time, in contrast to conventional machine learning and reinforcement learning methodologies (Rusek et al., 2020).

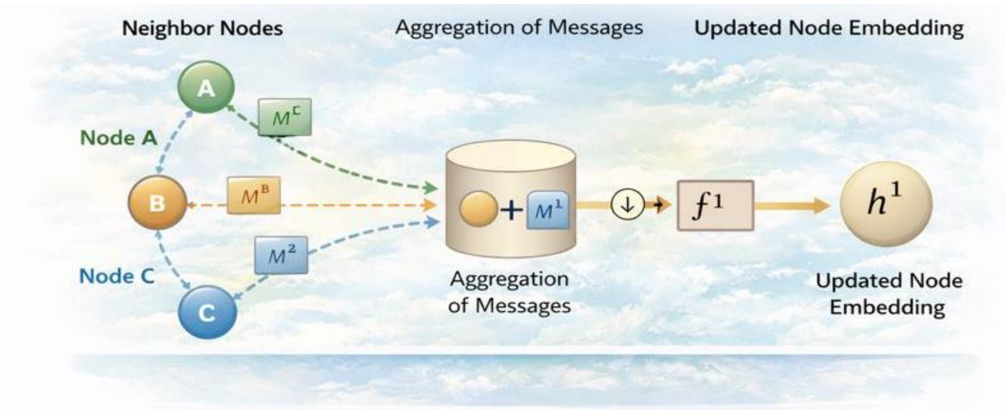

Fig. 5. Message-Passing Mechanism in a Graph Neural Network

Applications of GNNs in Satellite and Space Networks

In satellite and space networks, GNNs have been used to figure out how to route traffic between satellites, guess how the network will look, and choose paths that are aware of traffic. Their ability to understand how satellites are related to each other in space makes it possible to learn well across different constellation shapes and orbital arrangements. Recent research indicates that GNN-based routing models surpass topology-agnostic approaches in extensive LEO networks, yielding enhanced latency and stability amidst frequent topology alterations (Xu et al., 2024; Xiao et al., 2023).

Limitations of Pure GNN-Based Routing

Even though they have some benefits, pure GNN-based routing methods usually only work with snapshot-based inference and don't have clear ways to optimise decisions over time (Liu et al., 2023). Most GNN models work with static or discretised graph snapshots, which makes it hard to take into account the time-dependent and delayed rewards that are common in routing problems. Also, GNN inference by itself doesn't deal with trade-offs between exploration and exploitation or policy adaptation when traffic isn't stable. These constraints necessitate the amalgamation of GNNs with reinforcement learning to facilitate sequential, topology-aware routing decisions in dynamic STIN environments (Wu et al., 2021).

Hybrid GNN–Reinforcement Learning Frameworks for STIN Routing

Motivation for Hybrid GNN–RL Models

To route in STINs, you need to know about the topology and be able to make decisions in order. GNNs encode network structure and relational dependencies well, but they don't have ways to optimise and control things over time (Xu et al., 2024). RL, on the other hand, allows for policy optimisation by interacting with a changing environment. However, it doesn't work well when topology information isn't clearly modelled because it doesn't scale or generalise well. Rusek et al. (2020) emphasised that hybrid GNN–RL frameworks integrate these synergistic advantages by employing GNNs to produce topology-aware state representations that inform RL-based routing decisions, thus enhancing stability and scalability in dynamic STIN contexts, as illustrated in Fig 6.

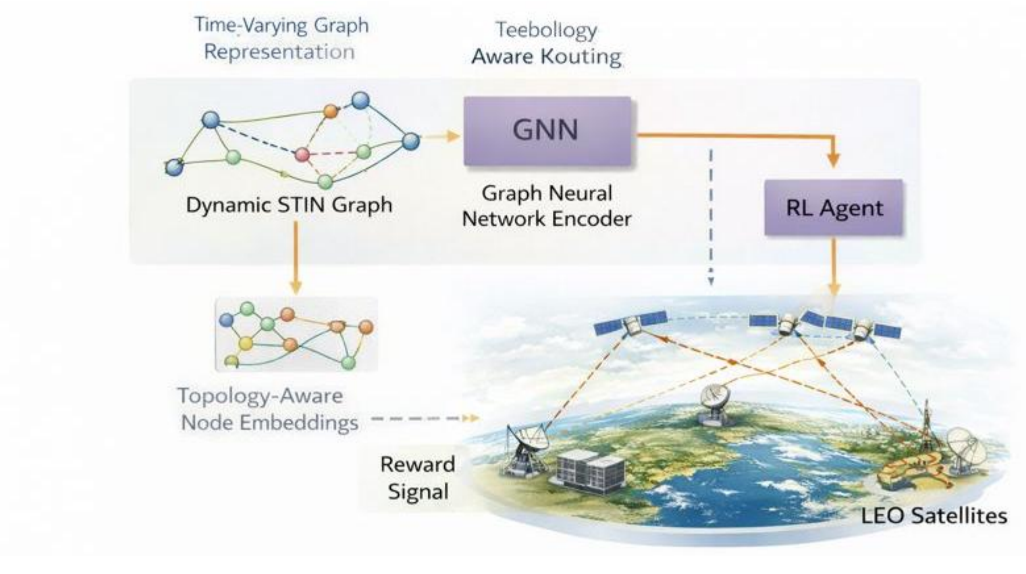

Fig. 6. Hybrid GNN–Reinforcement Learning Routing Framework for STINs

Architectural Patterns of GNN–RL Integration

Most GNN–RL routing frameworks use one of three architectural patterns. The most common one is where a GNN acts as a state encoder, turning the network graph into node or edge embeddings that an RL agent uses to choose an action. Other designs include training GNN and RL components together from start to finish or decentralised multi-agent architectures where each node follows a local GNN–RL policy based on partial observations. Zhang et al. (2023) Centralised training with distributed execution is often used to find a balance between training speed and deployment ease in large-scale STINs.

Representative Applications in STIN Routing

Hybrid GNN–RL techniques have been utilised for inter-satellite routing, cross-layer path selection, and traffic-aware routing within integrated satellite–terrestrial environments. Recent research indicates substantial performance improvements compared to traditional routing and independent reinforcement learning methods regarding end-to-end latency, packet delivery ratio, and convergence speed (Hu et al., 2024). GNN-based encoders allow policies trained on one constellation size or topology to work with new network configurations, which is very important for STIN to work in real life (Xu et al., 2024).

Comparative Perspective

Hybrid GNN-RL approaches converge faster, are more robust to changes in topology, and scale better as the size of the network grows than topology-agnostic RL approaches. Hybrid models outperform pure GNN inference in the long term by explicitly optimising routing policies over time. Liu et al. (2023) observed that these advantages are accompanied by heightened computational complexity and training overhead, underscoring the necessity for efficient model design and streamlined inference for onboard satellite deployment.

hierarchical and multi-agent extensions

Multi-Agent Learning in STIN Routing

Because STINs are so big and spread out, routing decisions are often made locally at satellites or gateways based on only a few observations of the network. This has led to the use of multi-agent reinforcement learning (MARL), in which several agents learn how to route data in a coordinated way by interacting with each other in a decentralised way. In STIN settings, MARL facilitates scalable routing management and diminishes reliance on global state distribution (Zhang et al., 2023). Nonetheless, non-stationarity resulting from simultaneous agent learning and restricted observability presents a considerable challenge, frequently resulting in unstable convergence and suboptimal coordination in the absence of explicit topology-aware representations (Roth et al., 2024).

Hierarchical STIN Architectures

As shown in Fig. 7, STINs have a hierarchical structure by nature. GEO, MEO, and LEO satellite layers work over different spatial extents and timescales, while terrestrial networks provide relatively stable backhaul connectivity (Giordani et al., 2020). Hierarchical routing architectures take advantage of this structure by splitting routing decisions across layers. This lets higher tiers do long-term planning while lower tiers do quick, local adjustments. Kodheli et al. (2021) say that this kind of decomposition makes calculations easier and makes systems more scalable, but it does require that lower-level topology and state information be properly abstracted to higher-level controllers.

Role of Topology-Aware Learning in Hierarchical Routing

Using GNNs for topology-aware learning is a natural way to create hierarchical abstraction in STIN routing. GNN-based embeddings can condense detailed topology data into small representations that help with decision-making at multiple levels (Liu et al., 2023). When combined with hierarchical or multi-agent RL, these representations make it possible to coordinate routing policies across satellite layers while still being able to grow and change. Even though the first results are promising, there are still open research problems with training complexity, cross-layer coordination, and real-time deployment (Xu et al., 2024).

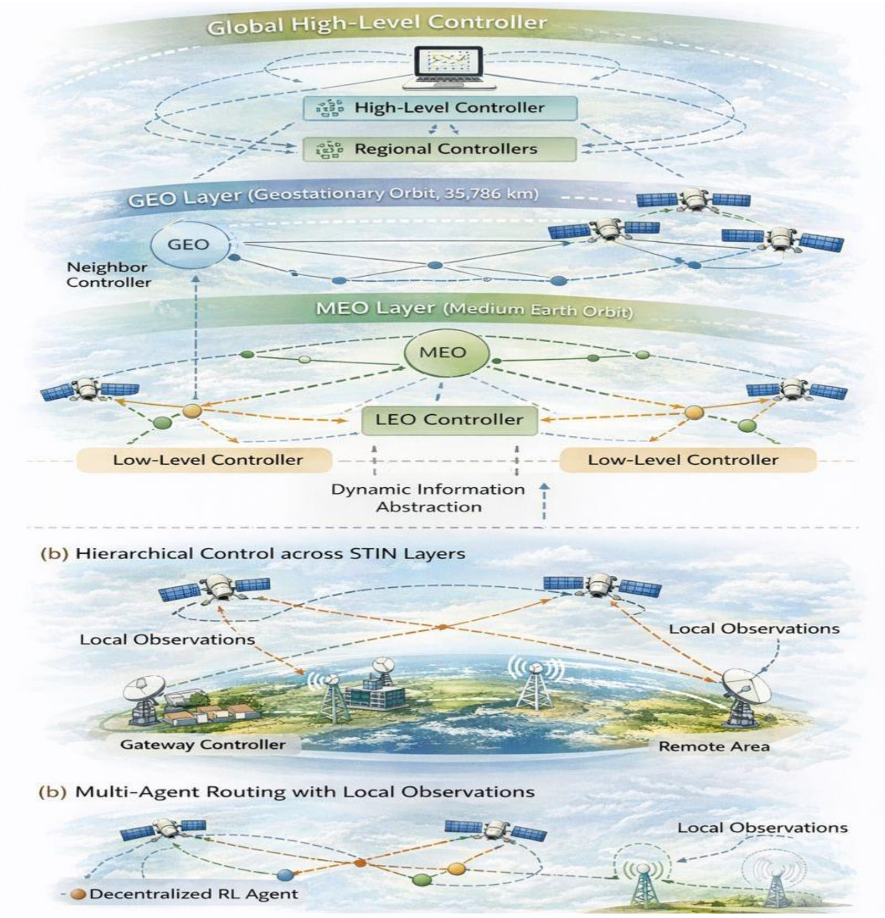

Fig. 7. Hierarchical and Multi-Agent Topology-Aware Routing in STINs

Datasets, Evaluation Methodologies, And Benchmarking Practices

Simulation-Based STIN Datasets

Because there isn't a lot of real-world measurement data, most studies on STIN routing use simulation-based datasets made with orbital mechanics models and network simulators. Some of the most common tools are STK-based orbital models, NS-3 satellite extensions, and custom graph generators that mimic changing inter-satellite and satellite-ground connectivity (Jiang et al. 2023). These datasets usually include models of satellite mobility, link intermittency, and traffic demand, but they often have very different constellation sizes, traffic assumptions, and update granularities, which makes it hard to compare studies.

Performance Metrics

Metrics like end-to-end latency, packet delivery ratio, throughput, and routing stability are often used to measure how well STINs route traffic (Goteti & Reddy, 2025). Learning-based studies also talk about how fast convergence happens, how rewards change over time, and how strong policies are when the topology changes. In satellite scenarios, energy use and control overhead are becoming more important because of limited resources on board. But the way metrics are defined and reported is very different, which makes it hard to compare routing methods objectively (Muzammal et al., 2022).

Benchmarking Challenges

Li et al. (2025) said that one of the biggest problems with testing topology-aware routing methods is that there aren't any standard benchmarks or reference datasets. Inconsistent performance claims are often caused by differences in constellation models, traffic patterns, and simulation assumptions. Additionally, numerous learning-based studies utilise oversimplified baselines that inadequately represent cutting-edge routing protocols (Veres, 2025). Setting up open datasets, shared evaluation scenarios, and reproducible benchmarks is still very important for moving forward with research on smart STIN routing (Mondragon Guadarrama et al., 2025).

A Comparative Quantitative Perspective

A significant limitation of the current literature is the scarcity of direct quantitative comparisons between topology-aware routing methods under identical conditions. To address this, Table I synthesizes representative performance results reported for GNN-based and GNN-RL approaches, highlighting the variability in gains over baseline methods (e.g., Dijkstra, Q-routing). While these figures demonstrate the potential of topology-aware learning, the diversity in simulation setups and performance metrics precludes a definitive ranking of methods. Future work should prioritize standardized evaluations to enable fair comparisons.

Table I: Illustrative Performance Gains of Topology-Aware Routing Methods

Approach	Baseline	Performance Gain	Network Scenario	Reference
GNN-based Routing	Dijkstra (Snapshot)	15-30% lower latency, 10-20% higher PDR	LEO Constellation (100 nodes)	(Xiao et al., 2023)
GNN-RL Hybrid (Central)	DQN (Flat-state)	25-40% faster convergence, 15% higher PDR	LEO Constellation (500 nodes)	(Xu et al., 2024)
GNN-RL Hybrid (Multi-agent)	Independent DQN	10-15% higher PDR, better load balancing	LEO-MEO Network (200 nodes)	(Hu et al., 2024)
GNN-based Federated RL	FedAvg with MLP	12-18% improvement in throughput	Multi-layer Satellite Network	(Li et al., 2025)

PDR: Packet Delivery Ratio; LEO: Low Earth Orbit; MEO: Medium Earth Orbit.

A Unified Evaluation Framework

To facilitate objective benchmarking, we propose a unified evaluation framework for topology-aware routing in STINs. This framework should comprise:

i. **Standardized Simulation Environment:** An open-source simulator (e.g., NS-3 with satellite modules, STK) with pre-defined, realistic constellation models (e.g., Starlink, OneWeb), traffic patterns, and link dynamics.

ii. **Core Performance Metrics:** Mandatory reporting of End-to-End Latency (mean, 95th percentile), Packet Delivery Ratio, and Routing Overhead.

iii. **Scenario-based Testing:** A set of canonical test scenarios that vary in scale (e.g., 50, 200, 1000 satellites), dynamics (slow vs. rapid topology change), and traffic load (low vs. high congestion).

iv. **Reproducibility:** A requirement for open-sourcing code and model configurations, a practice that is currently rare in the field.

Open Research Challenges and Future Directions

Even though there has been significant progress in topology-aware learning for STIN routing, some problems remain to be solved. One major problem is scaling up to mega-constellations, where thousands of satellites make high-dimensional graphs that change quickly (Chen et al. 2025). GNN-based models generalise more easily, but their high computational cost and message-passing overhead can make them hard to use in real time, especially on satellite platforms with limited resources. Creating lightweight, distributed learning architectures remains an area of research to be explored.

Another big problem is that learning environments are not always the same. In STINs, both the way the network is set up and the way traffic flows change over time, which go against the stationary assumptions that many reinforcement learning algorithms are based on. This frequently results in unstable training and diminished performance during deployment. Adding prediction-aware learning, continual learning, or meta-learning methods could make the system more stable in these situations (Alegre et al., 2021).

There is still not enough research on hierarchical and cross-layer coordination. Early research has examined hierarchical reinforcement learning and multi-agent frameworks; however, efficient coordination among satellite layers and between space and terrestrial segments remains an unresolved issue (Kolakowski et al., 2025). For scalable multi-layer routing control, you need topology-aware abstractions that find the right balance between decision granularity and computational efficiency.

The absence of standardised datasets and benchmarking frameworks continues to obstruct objective assessment and reproducibility. Setting up open STIN routing benchmarks that account for real-world mobility, traffic patterns, and failure scenarios is important for fair comparisons and the advancement of technology. Aligning with new 3GPP NTN standards will make future research even more useful (Chen et al., 2025).

Lastly, deployment limitations such as device processing limits, energy use, and communication overhead need to be carefully considered. One of the biggest problems is still figuring out how to translate simulation performance gains into real-world results. This requires working together to design learning algorithms and satellite system architectures.

Deployment Constraints and Hardware Limitations

Beyond algorithmic performance, the practical deployment of GNN-RL routing in STINs faces significant hardware and energy constraints. Onboard satellite processors are typically radiation-hardened and have limited computational and memory resources, making iterative message passing in GNNs and RL agent training prohibitively expensive. The primary challenges include:

i. **Computational Cost:** The inference time and energy consumption of a GNN layer grow with the node degree and model size, which is a critical bottleneck for high-speed, low-latency routing decisions.

ii. **Energy Efficiency:** The power budget for processing payloads is severely restricted. Complex learning models may compromise a satellite's primary mission functions.

iii. **Communication Overhead:** While GNNs reduce the need for global state exchange, they require periodic communication of node embeddings or model updates in distributed/multi-agent settings, which adds overhead to the already scarce ISL bandwidth. Bridging this gap will require the development of lightweight GNN architectures (e.g., via quantization, pruning, or knowledge distillation), efficient on-device inference engines, and a co-design approach that considers both algorithmic performance and hardware constraints from the outset.

CONCLUSION

This review has examined routing in STINs from a topology-aware learning perspective. It has also discussed the problems with classical and topology-agnostic learning-based methods in satellite-terrestrial environments that are highly dynamic and large-scale. In framing STINs as time-evolving graph systems, the review highlighted the need for routing decisions to explicitly account for structural dependencies, multi-hop interactions, and heterogeneous network layers. The survey revealed that although reinforcement learning offers adaptability to fluctuating network conditions, its efficacy is limited by flat state representations and inadequate generalisation. Graph Neural Networks get around these problems by letting you make structured, topology-aware representations that work with networks of different sizes and shapes. Hybrid GNN–RL frameworks emerge as a promising solution, combining relational awareness with long-term policy optimization and demonstrating superior performance in dynamic satellite and integrated network scenarios. The review also found major problems with scalability, non-stationarity, hierarchical coordination, benchmarking, and the practical implementation in the real world. To solve these problems, we need to advance lightweight graph learning, continuous and hierarchical learning systems, and standardised evaluation frameworks aligned with the new 6G and 3GPP NTN standards. In general, topology-aware learning is an important component for enabling intelligent routing in future STINs. This review aims to serve as a reference for researchers and practitioners working on scalable, robust, and deployable routing intelligence for next-generation satellite-terrestrial networks. It does this by bringing together current research trends and highlighting areas that need further work.

REFERENCES

1. Alegre, L., Bazzan, A., & da Silva, B. (2021). Quantifying the impact of non-stationarity in reinforcement learning-based traffic signal control. PeerJ Computer Science, 7, e575. https://doi.org/10.7717/peerj-cs.575
2. Chen, N., Song, Y., Cao, Y., Sun, Z., Zhao, B., Wang, M., He, D., & Peng, G. (2025). Network-layer perspectives on satellite–terrestrial integrated networks in 6G: A comprehensive review. Engineering. https://doi.org/10.1016/j.eng.2025.05.012
3. Chen, Y., Cao, H., Wang, L., Chen, D., Liu, Z., Zhou, Y., & Shi, J. (2025). Deep Reinforcement Learning-Based Routing Method for Low Earth Orbit Mega-Constellation Satellite Networks with Service Function Constraints. Sensors, 25(4), 1232. https://doi.org/10.3390/s25041232
4. Ekici, E., Akyildiz, I. F., & Ergin, D. (2022). Time-dependent routing in satellite networks. IEEE/ACM Transactions on Networking, 30(5), 2134–2147.
5. Ekici, E., Akyildiz, I. F., & Ergin, D. (2022). Time-dependent routing in satellite networks. IEEE/ACM Transactions on Networking, 30(5), 2134–2147.
6. Giordani, M., Polese, M., Mezzavilla, M., Rangan, S., & Zorzi, M. (2020). Toward 6G networks: Use cases and technologies. IEEE Communications Magazine, 58(3), 55–61.
7. Goteti, D., & Reddy, V. (2025). AI-driven routing pipeline in software-defined networks using DQL: A mini review. Frontiers in Artificial Intelligence, 8, 1685155. https://doi.org/10.3389/frai.2025.1685155.
8. Hu, Z., Han, C., Gerstacker, W., & Akyildiz, I. F. (2024). Tera-SpaceCom: GNN-based deep reinforcement learning for joint resource allocation and task offloading in terahertz-band space networks. arXiv preprint. https://doi.org/10.48550/arXiv.2409.07911
9. Jiang, W., Zhan, Y., Xiao, X., & Sha, G. (2023). Network simulators for satellite-terrestrial integrated networks: A survey. IEEE Access, 11, 1–1. https://doi.org/10.1109/ACCESS.2023.3313229
10. Kodheli, O., Lagunas, E., Maturo, N., Sharma, S. K., Shankar, B., Montoya, J. F. M., & Goussetis, G. (2021). Satellite communications in the new space era: A survey and future challenges. IEEE Communications Surveys & Tutorials, 23(1), 70–109.
11. Kołakowski, R., Tomaszewski, L., Tępiński, R., & Kukliński, S. (2025). Hierarchical traffic engineering in 3D networks using QoS-aware graph-based deep reinforcement learning. Electronics, 14. https://doi.org/10.3390/electronics14051045
    Li, R., Zhu, J., Mao, Y., Shi, Y., Wang, T., & Jiang, C. (2025). Topology-aware routing for federated learning over multi-layer satellite networks. IEEE Wireless Communications and Networking Conference (WCNC), 1–6. https://doi.org/10.1109/WCNC61545.2025.10978815
12. Liu, X., Han, C., Wang, J., & Akyildiz, I. F. (2023). Deep learning for non-terrestrial networks: A survey. IEEE Communications Surveys & Tutorials, 25(3), 1890–1931.
13. Mondragon Guadarrama, J., Branco, P., Jourdan, G.-V., Gutiérrez-Rodríguez, A., & Biswal, R. (2025). Advanced IDS: A comparative study of datasets and machine learning algorithms for network flow-based intrusion detection systems. Applied Intelligence, 55. https://doi.org/10.1007/s10489-025-06422-4

14. Muzammal, S. M., Murugesan, R. K., Jhanjhi, N., Hossain, M. S., & Yassine, A. (2022). Trust and Mobility-Based Protocol for Secure Routing in Internet of Things. Sensors, 22(16), 6215. https://doi.org/10.3390/s22166215
15. Roth, M. M. H., et al. (2024). Shaping rewards, shaping routes: On multi-agent deep Q-networks for routing in satellite constellation networks. DLR Technical Report.
16. Rusek, K., Suárez-Varela, J., Mestres, A., Barlet-Ros, P., & Cabellos-Aparicio, A. (2020). Unveiling the potential of graph neural networks for network modeling and optimization. IEEE/ACM Transactions on Networking, 28(6), 2249–2262.
17. Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. IEEE Transactions on Neural Networks, 20(1), 61–80.
18. Veres, P. (2025). ML and Statistics-Driven Route Planning: Effective Solutions Without Maps. Logistics, 9(3), 124. https://doi.org/10.3390/logistics9030124
19. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 4–24.
20. Xiao, Z., Liu, J., Zhang, Q., & Wang, S. (2023). GNN-based routing for dynamic satellite networks. IEEE Network, 37(6), 134–141.
21. Xu, P., Feng, M., Zhou, J., Xiao, P., & Jiang, T. (2024). Inter-satellite routing for LEO satellite networks: A GNN and DRL integrated approach. In Proceedings of the IEEE/CIC International Conference on Communications in China (ICCC).
22. Zhang, Q., Wang, S., Li, Y., & Zhang, H. (2023). Learning-based routing in dynamic satellite networks: Challenges and opportunities. IEEE Network, 37(5), 122–129.
23. Zhu, X., et al. (2021). Reinforcement learning-based routing algorithm in satellite–terrestrial integrated networks. Conference proceedings.