

Review

Graph Neural Network-Enabled Intelligence for Unmanned Aerial Vehicle Systems: A Comprehensive Review

Rinkuben Patel  and Areej Salaymeh * 

College of Business and Information Technology, Lawrence Technological University, 21000 W Ten Mile Rd, Southfield, MI 48075, USA; rpatel3@ltu.edu

* Correspondence: asalaymeh@ltu.edu

Highlights

What are the main findings?

- Six GNN architecture categories for UAVs: Analysis of 196 papers identifies six different GNN architectures (convolutional, attentional, sampling-based, spatio-temporal, distributed, and resource-efficient), which have different approaches to scalability, topology modification, and computational limitations. No one type is best suited for all UAV tasks.
- Simulation does not always predict deployment success: across trajectory planning, cooperative tracking, FANET routing, and traffic prediction, simulation results do not always reliably reflect real-world performance, and few studies evaluate the same method in both settings. Hybrid methods combining GNN learning with classical optimization often outperform purely learned approaches.

What are the implications of the main findings?

- Architecture should be designed to suit the objective rather than the method. Use GCN or GraphSAGE for large swarms. For tasks with variable sensor quality, use attention-based models. For drones with limited battery and memory, use compressed variants. This practical mapping saves practitioners from reading all 196 papers to make an informed decision.
- Four research priorities emerge from deployment gaps: (1) Design lightweight GNNs that adapt to network topology changes while respecting energy budgets; (2) Build distributed learning systems that reduce communication overhead in large swarms; (3) Test learned policies on actual hardware, not just in simulation; (4) Combine learned policies with formal safety rules to guarantee collision avoidance and other constraints.

Abstract

Coordinating multiple unmanned aerial vehicles (UAVs) at scale remains challenging through centralized control or fixed rule sets, particularly when vehicles must operate under intermittent communication links, incomplete observability, and constrained onboard computational resources. Graph Neural Networks (GNNs) have emerged as a promising framework for addressing these challenges; however, existing surveys do not systematically relate GNN architectural decisions to the operational constraints imposed by UAV platforms during deployment. This survey reviews 196 scholarly studies published between 1987 and 2026 to develop such a framework. A GNN architecture and deployment taxonomy is organized into six major categories—Convolutional, Attentional, Sampling-Based, Spatio-Temporal, Distributed, and Resource-Efficient—each examined through dedicated architectural subsections and evaluated in the context of UAV system constraints. Four primary application domains are examined: multi-UAV trajectory planning, cooperative



Academic Editor: Carlos Tavares Calafate

Received: 23 May 2026

Revised: 8 July 2026

Accepted: 14 July 2026

Published: 18 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

target tracking, communication-aware network optimization in Flying Ad Hoc Network (FANET) environments, and spatio-temporal airspace traffic prediction. Within these domains, the analysis highlights how architectural choices influence scalability, adaptability to dynamic conditions, and computational efficiency. Several deployment challenges consistently emerge, including maintaining tractable inference as swarm size increases, adapting graph representations under high mobility, and operating within the limitations of onboard computational resources. Based on these findings, a set of architecture-selection guidelines is derived to support deployment under varying operational conditions. Emerging research directions are also discussed, particularly the integration of GNNs with reinforcement learning, federated edge computing, and next-generation wireless communication systems. Overall, this survey bridges the gap between methodological development and practical deployment, providing a structured foundation for evaluating GNN suitability in real-world multi-UAV environments.

Keywords: Graph Neural Networks; Unmanned Aerial Vehicles; multi-UAV systems; swarm intelligence; trajectory planning; target tracking; Flying Ad Hoc Network (FANET); edge computing; Sixth-Generation (6G) networks; distributed learning

1. Introduction

Recent advances in unmanned aerial vehicle (UAV) technologies have significantly expanded their role in applications such as disaster management, infrastructure monitoring, telecommunications, and environmental observation [1–3]. As the applications of UAVs grow from single-UAV missions to large-scale multi-UAV operations and autonomous swarm systems, their operational complexity has increased significantly [4–7]. Conventional control paradigms based on centralized architectures and handcrafted solutions are often insufficient to manage the dynamic, uncertain, and highly interactive characteristics of such systems [8]. These challenges have led to increased interest in data-driven learning approaches capable of modeling complex spatial relationships, temporal dependencies, and multi-agent interactions.

Graph Neural Networks provide a suitable learning framework for modeling relational and networked structures of multi-UAV systems [9–12]. GNNs naturally represent UAV networks as dynamic graphs, where nodes correspond to individual UAV agents and edges represent communication links, spatial proximity, or mission-relevant interactions. This characteristic differentiates GNNs from conventional deep learning models, which typically operate on data with fixed-dimensional Euclidean structures [13,14]. By leveraging graph-based representations, learning algorithms can exploit the intrinsic structure of communication and coordination networks, thereby facilitating enhanced scalability, adaptability, and generalization in tasks such as resource allocation, communication optimization, and cooperative decision-making [15–17]. Building on these graph-based learning principles, recent studies have further explored GNN-based approaches for UAV trajectory planning, cooperative navigation, and target tracking within intricate multi-agent environments.

1.1. Why Graph-Based Learning and GNNs for UAV Systems?

A fundamental question arises: why are graph-based learning and GNN architectures preferable to other machine learning paradigms for UAV systems? The answer emerges from examining both the relational structure of UAV systems and the fundamental constraints governing their operation. UAV systems are subject to several system-level limitations that strongly influence the design of learning and control mechanisms [1,18]. These constraints include onboard computation and energy limitations, which restrict algorithmic

complexity on individual UAV platforms [19,20]. High mobility and FANET dynamics result in intermittent and unreliable connectivity, posing challenges for stable communication and coordination [21–24]. Partial observability of the environment and neighboring agents further necessitates decentralized decision-making under uncertainty [25,26]. In large-scale swarm deployments, decentralized execution requirements make persistent centralized coordination impractical [8,27,28]. In addition, safety-critical applications such as collision avoidance, formation control, and Ultra-Reliable Low-Latency Communication (URLLC)-driven missions impose strict latency constraints [29–32]. These constraints prevent the effective deployment of centralized learning and global optimization approaches, particularly in large-scale or mission-critical UAV swarm environments.

To provide a unified conceptual perspective, Figure 1 summarizes the constraint-driven relationship between UAV system limitations, GNN architectural choices, and intelligent multi-UAV outcomes.

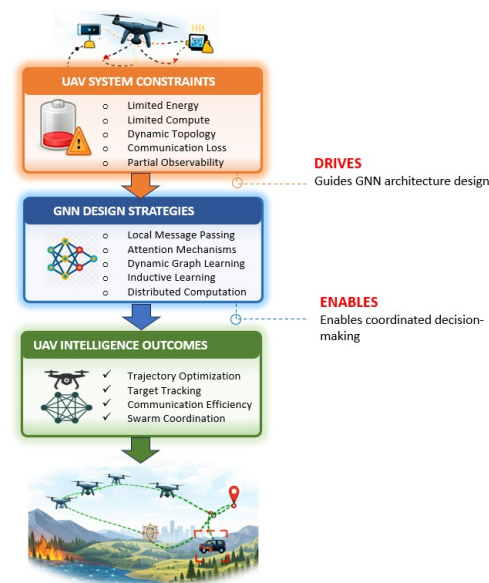


Figure 1. Constraint-driven design of graph neural network architectures for intelligent multi-UAV systems. UAV system constraints guide the selection of GNN architectural and deployment choices, which in turn enable coordinated decision-making and task-specific intelligence outcomes.

The framework highlights how UAV system constraints, GNN architectural choices, and multi-UAV capabilities are interconnected. Constraints such as limited onboard energy and computational resources encourage lightweight and localized message-passing mechanisms, while dynamic network topology and intermittent connectivity require adaptive architectures, including attention-based and dynamic graph models. Partial observability and decentralized execution further motivate distributed GNN frameworks that support localized decision-making without reliance on global information.

These design choices collectively enhance trajectory optimization, cooperative coordination, communication efficiency, and adaptive sensing. Overall, Figure 1 highlights that GNN architecture selection should be driven by system-level constraints rather than model popularity, thereby enabling more reliable and scalable UAV intelligence.

Graph Neural Networks provide an effective framework to address these challenges through several key characteristics. GNNs introduce a structural inductive bias, which allows models to learn from graph-structured data through message passing that captures relational dependencies without requiring complete information about the network topology [14,33]. In addition, GNN architectures exhibit permutation equivariance at the node level, meaning that if the ordering of UAV agents changes, the corresponding outputs

are permuted consistently. This property ensures stable node-level predictions and enables GNNs to generalize across UAV swarms of varying sizes, supporting scalable and decentralized coordination [27,28]. The message-passing mechanism also supports distributed computation, enabling decentralized implementations aligned with the operational constraints of multi-UAV systems [34–37].

Relational structures of this type arise across a wide range of UAV applications. In trajectory planning, spatial proximity graphs capture collision avoidance and coordinated path formation [38–41]. Cooperative target tracking relies on observation graphs to model sensor coverage and inter-agent dependencies for distributed estimation [42–45]. Communication networks are commonly represented using mesh topologies, where connectivity must be optimized for latency, reliability, and throughput [26,46–48]. In spatio-temporal air traffic forecasting, historical trajectory data forms temporal graphs that capture evolving flow patterns and airspace congestion [49–51]. In contrast, conventional Convolutional Neural Networks (CNNs) designed for grid-structured data and Recurrent Neural Networks (RNNs) for sequential data are not well suited to capture these complex and heterogeneous graph structures.

1.2. Motivation and Research Gap

Despite significant progress in both UAV autonomy and graph neural network research, these areas are often studied in isolation. Surveys centered on UAVs [1–3,5] provide extensive analyses of hardware platforms, control architectures, and application domains, but they do not adequately address contemporary learning-based methodologies. On the other hand, surveys on Graph Neural Networks [9,12,13,52] investigate architectural innovations, theoretical foundations, and general applications in detail, but they rarely consider the specific requirements, constraints, and dynamics of aerial robotic systems.

This survey directly connects UAV system constraints—such as high mobility, FANET dynamics, and limited onboard energy—and graph learning design principles. Previous surveys analyze either UAV autonomy frameworks or GNN architectures in isolation. In contrast, this unified perspective facilitates principled model selection for practical UAV implementations.

To address this gap, this survey focuses on three fundamental research questions:

1. How do architectural choices in Graph Neural Networks align with the requirements of specific UAV applications, such as trajectory planning, target tracking, and communication optimization?
2. What challenges arise when deploying GNN-based solutions on resource-constrained platforms, particularly with respect to computational complexity and energy-efficient inference?
3. How can emerging technologies be integrated with graph-based learning to enable next-generation intelligent UAV systems?

Building on these research questions, the contribution of this survey can be positioned as follows.

Positioning of This Review: Unlike purely systematic or bibliometric surveys, this work adopts a design-oriented and integrative perspective. Rather than only summarizing existing studies, it synthesizes insights across the literature to derive unified design principles, analyze architectural trade-offs, and evaluate deployment feasibility. This approach provides actionable guidance for selecting and adapting graph neural network models in real-world UAV systems operating under practical constraints. Consequently, this survey serves both as a reference for researchers and as a practical guide for designing next-generation intelligent UAV systems.

1.3. Comparison with Existing Surveys

To clearly position this survey with respect to existing literature, a structured comparison of prior surveys is presented in Table 1. Existing works can be broadly categorized into two groups: (i) UAV-focused surveys and (ii) Graph Neural Network (GNN)-focused surveys.

UAV-centric surveys, including Mozaffari et al. [1], Chung et al. [6], and Shakhathreh et al. [2], provide comprehensive discussions on communication architectures, swarm coordination, and application domains. However, these works primarily focus on system design and control strategies, with limited consideration of modern learning-based approaches, particularly graph-based learning methods.

On the other hand, surveys on Graph Neural Networks, including Wu et al. [9] and Zhou et al. [13], offer in-depth coverage of GNN architectures, theoretical foundations, and applications across various domains. Despite their breadth, these studies do not address the unique challenges of UAV systems, such as dynamic topology, communication constraints, and decentralized decision-making.

More recent works, such as Kose et al. [19], explore efficient and resource-constrained GNN architectures. While these contributions are relevant to embedded systems, they are not specifically tailored to UAV environments and do not provide application-level insights.

In contrast, this survey bridges the gap between these two research domains by providing a unified analysis of GNN architectures within the context of UAV systems. Unlike prior work, it integrates model design, application requirements, and deployment constraints into a single framework, enabling a more systematic understanding of how graph-based learning can be effectively applied to UAV intelligence.

Overall, Table 1 highlights that existing surveys either focus on UAV systems without leveraging advanced learning models or explore GNN methodologies without addressing UAV-specific challenges. This work uniquely combines both perspectives, offering a comprehensive and application-driven analysis.

Concretely, none of the surveys in Table 1 maps GNN architectural choices to UAV operational constraints. The UAV-focused surveys [1,2,6] contain no analysis of GNN architectures, and the GNN-focused surveys [9,13,19] contain no analysis of UAV constraints. This is visible in the “UAV Constraints” and “Cross-domain” columns of Table 1, where every prior row is marked “No.” The specific contribution claimed here, a constraint-to-architecture mapping with deployment guidance, is therefore absent from each prior survey, and the remaining sections develop it.

Table 1. Comparison of existing UAV and GNN surveys with the proposed work.

Survey	Focus	UAV Systems	GNN Models	Graph-Based Learning	UAV Constraints	Cross-Domain	Quantitative	Key Limitation
Mozaffari et al. [1]	UAV	Yes	No	Limited	Yes	No	No	Limited integration of learning-based methods
Shakhathreh et al. [2]	UAV	Yes	No	Limited	Yes	No	No	Primarily focused on system-level design
Chung et al. [6]	UAV	Yes	No	Limited	Yes	No	No	Limited incorporation of AI/ML techniques
Wu et al. [9]	GNN	No	Yes	Yes	No	No	No	Not tailored to UAV systems
Zhou et al. [13]	GNN	No	Yes	Yes	No	No	No	Limited application-level evaluation
Kose et al. [19]	GNN	No	Yes	Yes	No	No	No	Not tailored to UAV-specific deployment
This Work	Hybrid (UAV + GNN)	Yes	Yes	Yes	Yes	Yes	Yes	Unified and application-driven analysis

1.4. Contributions and Scope

This survey establishes a unified perspective on GNN architectures and deployment-oriented categories for UAV systems by systematically analyzing how graph-based learning addresses autonomous aerial intelligence challenges. The main contributions include:

- Unified architecture and deployment taxonomy for GNN-enabled UAV systems: A structured taxonomy categorizes major GNN variants, including Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), GraphSAGE, and spatio-temporal GNNs. Each category is analyzed in terms of its suitability for UAV environments, highlighting how architectural design choices align with system-level requirements such as scalability, adaptability, and robustness.
- Application-driven analysis across UAV domains: GNN-based approaches are systematically examined across key UAV applications, including multi-UAV trajectory planning and collision avoidance, cooperative target tracking and surveillance, communication-aware modeling, and spatio-temporal prediction. This analysis provides insights into how graph-based representations enable efficient coordination and decision-making in dynamic environments.
- Critical evaluation of deployment challenges: This work goes beyond model-level analysis by identifying practical challenges associated with real-world deployment. Key issues such as scalability to large swarms, dynamic graph updates under high mobility, communication constraints, and limited onboard computational resources are critically examined, along with existing mitigation strategies.
- Integration with emerging technologies: The survey explores the intersection of GNNs with emerging paradigms, including reinforcement learning, federated learning, and edge-computing frameworks. These integrations are discussed in the context of enabling adaptive, decentralized, and resource-efficient UAV intelligence.
- Identification of research gaps and future directions: Open challenges are highlighted, including the simulation-to-real gap, lack of standardized benchmarks, and absence of formal safety guarantees. Based on these observations, future research directions are outlined to guide the development of reliable and scalable GNN-based UAV systems.

Overall, this survey provides a comprehensive and application-oriented framework that connects GNN methodologies with UAV system design, offering practical insights for both researchers and practitioners.

1.5. Paper Selection Methodology

A structured and systematic literature review methodology is adopted to ensure comprehensive coverage and reproducibility. The study focuses on Graph Neural Network (GNN) applications in UAV systems, particularly in multi-UAV collaboration, trajectory optimization, target tracking, and communication-aware networks.

A comprehensive search was conducted across major academic databases, including IEEE Xplore, Elsevier ScienceDirect, SpringerLink, MDPI, ACM Digital Library, arXiv, and Google Scholar. The search covered publications from 1987 to 2026.

The final database search was executed in May 2026, and the most recent included records were published in March 2026. Within each database, the keyword groups listed in this section were combined using the Boolean operator AND across groups and OR within each group. Searches were limited to records published between 1987 and 2026 and to English-language documents. Representative search queries adapted to the syntax of each database are provided in Appendix A.

The lower bound of 1987 reflects the inclusion of foundational swarm-behavior work that predates graph neural networks, particularly Reynolds' distributed flocking model [53], which established the local-interaction principles that graph-based learning later formal-

ized. Graph neural networks themselves emerged later, with early formulations by Gori et al. [54] and Scarselli et al. [12]. These foundational references are included for historical context only; The GNN-for-UAV literature analyzed in this survey is concentrated primarily after 2018.

The following keyword combinations were used:

- “Graph Neural Network”, “Graph Convolutional Network”, “GNN”, “GCN”, “Graph Attention Network”, “GAT”, “GraphSAGE”, “Spatio-Temporal Graph Neural Network”, “Dynamic Graph Neural Network”.
- “UAV”, “Unmanned Aerial Vehicle”, “Drone”, “Multi-UAV”, “UAV Swarm”, “Flying Ad Hoc Network”, “FANET”.
- “Trajectory Planning”, “Path Planning”, “Collision Avoidance”, “Target Tracking”, “Cooperative Tracking”, “Communication Optimization”, “Routing”, “Resource Allocation”, “Task Allocation”, “Spatio-Temporal Prediction”.
- “Multi-Agent Reinforcement Learning”, “MARL”, “Graph Reinforcement Learning”, “Decentralized Coordination”, “Swarm Coordination”, “Federated Learning”, “Edge Computing”, “Mobile Edge Computing”, “6G”, “Digital Twin”.
- Combined terms: “GNN UAV”, “Graph neural network drone”, “Multi-agent UAV graph”, “Graph-based UAV coordination”, “GNN-based UAV communication”, “GNN UAV trajectory planning”, “GNN FANET routing”, “Graph learning UAV swarm”.

The initial database search identified 525 records across multiple academic sources. After removing 49 duplicate records, 476 studies remained for title and abstract screening. Following the first-stage screening, 294 full-text articles were assessed for eligibility. After full-text review and the application of inclusion and exclusion criteria, 196 eligible studies were selected for the final qualitative synthesis.

In addition to the 196 systematically reviewed studies, a small number of foundational and architecture-background references are cited for contextual and methodological completeness.

Inclusion criteria:

- GNN or graph-based learning applied to UAV systems;
- Multi-agent coordination, communication, or control problems;
- Experimental validation or theoretical contribution.

Exclusion criteria:

- Non-peer-reviewed articles, except highly relevant preprints used for emerging or very recent research directions;
- Papers without methodological contribution;
- Duplicate or redundant studies.

In the final exclusion stage, nine studies were classified as position or weak-evidence papers. These were defined as works that present claims or recommendations without experimental validation, theoretical proof, or methodological contribution directly applicable to GNN-based UAV systems.

This structured process ensures that the review is comprehensive, reproducible, and focused on studies directly relevant to GNN-enabled UAV systems.

The PRISMA workflow shown in Figure 2 summarizes the study selection process, including database search, duplicate removal, screening, eligibility assessment, and final inclusion.

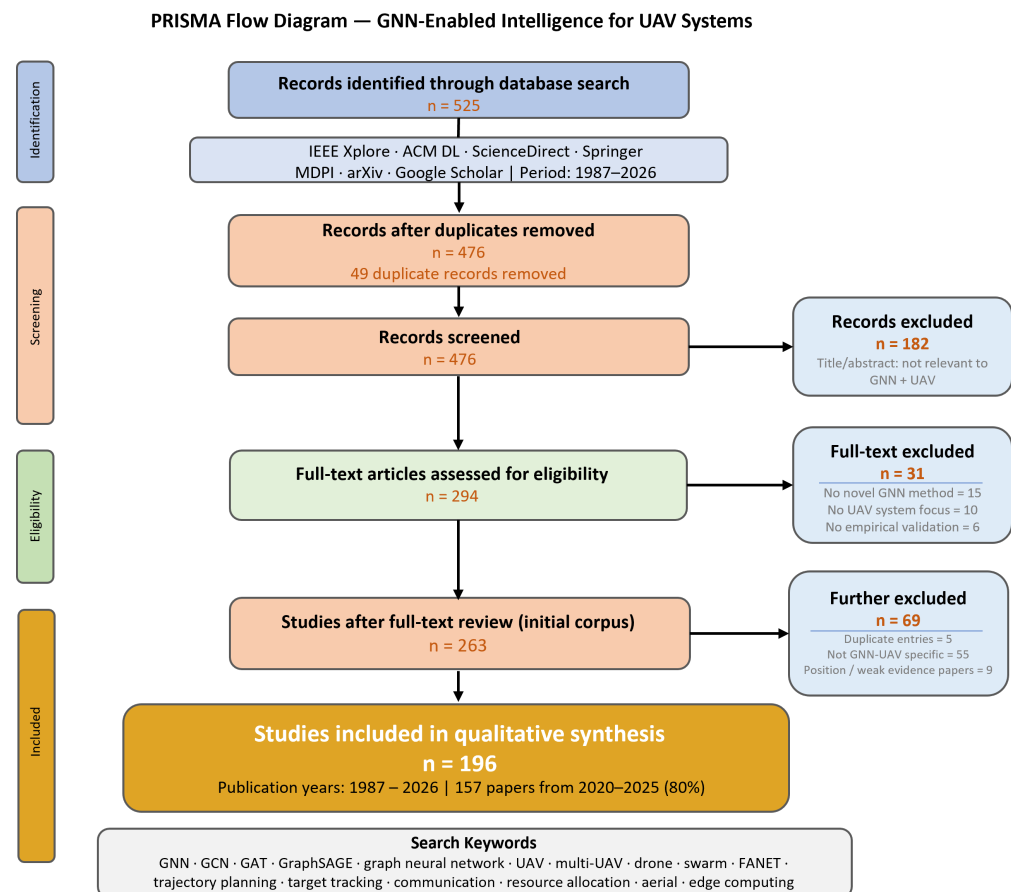


Figure 2. PRISMA flow diagram illustrating the study selection process for GNN-enabled UAV systems.

Figure 3 illustrates the temporal distribution of the reviewed publications, highlighting the rapid growth of GNN-enabled UAV research after 2018.

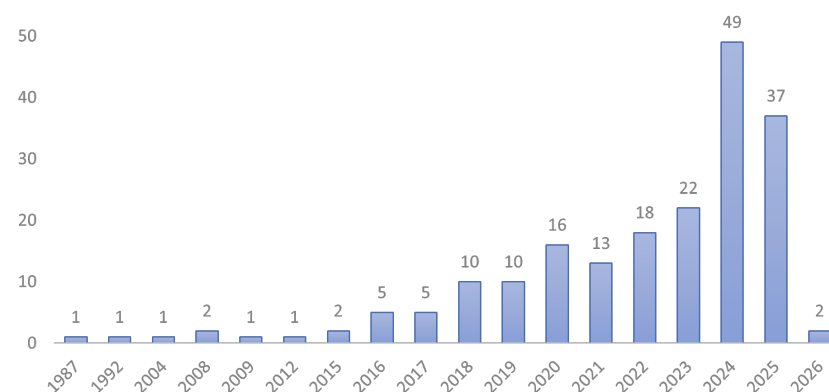


Figure 3. Temporal distribution of the reviewed publications.

Temporal Distribution

A total of 56% of the reviewed papers were published between 2023 and 2026, indicating rapid growth in research interest in GNN-enabled UAV systems.

This temporal trend indicates that GNN-enabled UAV intelligence is a rapidly emerging research area rather than a fully mature field. The sharp increase after 2020 reflects the growing convergence of graph learning, autonomous aerial systems, and multi-agent coordination. At the same time, the concentration of recent studies suggests that the literature remains fragmented, with limited standardization in benchmarks, evaluation settings, and

deployment conditions. Moreover, many recent contributions are still validated primarily in simulation, indicating that real-world generalization and reproducibility remain open challenges. These observations underscore the need for more unified evaluation protocols and stronger experimental validation in future work.

1.6. Organization

The remaining part of this survey is structured as follows. Section 2 discusses the relevant background of the GNN architecture and deployment taxonomy, including Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), GraphSAGE, spatio-temporal GNNs, distributed GNNs, and resource-efficient GNNs, together with UAV system models. Section 3 presents a comprehensive overview of the applications of GNNs in multi-UAV trajectory design, cooperative target tracking, communication networks, and air traffic prediction. Section 4 discusses the limitations of current GNN-based UAV approaches, highlighting gaps between simulation-based research and real-world deployment. Section 5 examines the core technical challenges involved in deployment, with particular emphasis on scalability, dynamic graph structures, and computational constraints. Section 6 outlines potential research directions, focusing on reinforcement learning, edge computing, and the integration of future 6G communication networks. Finally, Section 7 concludes the paper with a synthesis of key findings and practical architecture-selection guidance.

2. Background

2.1. Graph Neural Network Architecture and Deployment Taxonomy for UAV Systems

The taxonomy used here classifies GNN approaches along two orthogonal axes. The first axis is the message-passing mechanism, which defines how a model aggregates neighbor information: fixed convolutional weighting (GCN), learned attention (GAT), sampled aggregation (GraphSAGE), or temporally coupled aggregation (spatio-temporal GNNs). The second axis is the deployment paradigm, which defines how computation is distributed and compressed for hardware: distributed execution and resource-efficient compression. The two axes are independent: the first is mutually exclusive at the level of the core operator, whereas the second is not, because any operator can be distributed or compressed. The first axis therefore determines what a model can represent, and the second determines whether it can run on a UAV.

Each formulation below is interpreted through the deployment property implied by its core operator. GCN emphasizes normalized local aggregation, which supports scalable neighbor-based coordination; GAT introduces adaptive neighbor weighting, which is particularly relevant under heterogeneous sensing and variable link quality; and GraphSAGE employs sampled inductive aggregation, which supports UAV swarms with dynamically changing membership. Table 2 summarizes these correspondences and their deployment implications.

Table 2. Analytical relevance of standard GNN formulations to UAV deployment constraints.

Architecture	Key Mathematical Mechanism	UAV-System Interpretation	Deployment Implication
GCN	Normalized neighborhood aggregation	UAVs exchange local state information with nearby agents	Suitable for scalable swarm coordination, but node representations are prone to over-smoothing as message-passing depth increases
GAT	Learnable attention coefficient α_{ij}	UAVs weight neighboring agents according to link quality, sensing relevance, or mission context	Useful for interference-limited or heterogeneous UAV networks, but incurs higher per-edge computational cost
GraphSAGE	Sampled inductive aggregation	UAVs aggregate information from sampled neighbors and generalize to unseen agents	Suitable for dynamic swarms with node arrival, failure, or topology changes, though neighbor sampling introduces approximation in the aggregated representation

2.1.1. Graph Convolutional Networks

Graph Convolutional Networks perform feature aggregation through neighborhood convolution operations. The layer-wise propagation rule is defined as [14]:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}\right) \quad (1)$$

where $\tilde{A}$ is the adjacency matrix with added self-loops and $\tilde{D}$ is its corresponding degree matrix, defined as

$$\tilde{A} = A + I_N, \quad \tilde{D}_{ii} = \sum_j \tilde{A}_{ij}. \quad (2)$$

where $W^{(l)}$ is the layer weight matrix, σ is a nonlinearity such as ReLU, and $H^{(0)} = X$ [14].

Mapping to UAV Constraints: In multi-UAV systems, GCN message-passing aligns with localized interaction constraints arising from limited communication range and energy budgets [34,36]. Each UAV aggregates neighbor information without global knowledge, enabling scalable coordination. Symmetric normalization ensures balanced flow across heterogeneous networks [27,28]. For trajectory planning, GCNs process relative positions, enabling collision-free generation [41,55]. Permutation equivariance ensures consistent node-level policy behavior under node reordering, which is critical for UAV swarm coordination [56].

2.1.2. Graph Attention Networks

Graph Attention Networks (GATs) [57] and their heterogeneous extensions introduce learnable attention coefficients that weight neighbor contributions [58]. The attention mechanism computes:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a^T [Wh_i \| Wh_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(a^T [Wh_i \| Wh_k]))} \quad (3)$$

where a is a learnable attention vector and $\|$ denotes concatenation. The output features are computed as

$$h'_i = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} Wh_j\right) \quad (4)$$

Mapping to UAV Constraints: Attention mechanisms enable dynamic neighbor prioritization, which is essential under interference-limited FANET conditions [21–23]. Higher weights assigned to UAVs with stronger signals or relevant information facilitate interference mitigation and selective sharing [15,16,47]. Multi-head attention learns diverse patterns for spatial proximity, communication quality, and mission relevance [57]. Heterogeneous GATs suit UAV systems integrating communication states, sensors, and mission parameters [58]. Heterogeneous Graph Attention Networks (HGATs) model UAV-UAV and UAV-sensor links with distinct mechanisms. Learning-based approaches have also been explored for efficient coordination and data collection in UAV-enabled IoT systems [59].

2.1.3. Inductive Learning with GraphSAGE

GraphSAGE (Sample and aggreGatE) enables inductive learning by sampling neighborhoods and aggregating features through learnable functions [33]:

$$h_v^k = \sigma\left(W^k \cdot \text{CONCAT}\left(h_v^{k-1}, \text{AGGREGATE}\left(\{h_u^{k-1}, \forall u \in \mathcal{N}(v)\}\right)\right)\right) \quad (5)$$

where CONCAT denotes concatenation or element-wise operations, and AGGREGATE can be mean, LSTM, or pooling functions. Unlike transductive GCNs requiring the entire graph during training, GraphSAGE generates embeddings for unseen nodes through local neighborhood sampling.

Mapping to UAV Constraints: GraphSAGE's inductive capability is critical for UAV swarms with time-varying agent populations due to arrivals, departures, or failures [27,28].

Key advantages:

- Zero-shot generalization: This learns aggregation functions rather than node-specific embeddings, enabling new UAVs to join swarms without retraining [60].
- Computational efficiency: Neighborhood sampling reduces computational complexity by limiting the number of neighbors aggregated at each layer, enabling scalable and efficient inference on resource-constrained platforms [33].

2.1.4. Spatio-Temporal Graph Neural Networks

Spatio-temporal GNNs model spatial dependencies through graph structure and temporal dynamics through recurrent, convolutional, or attention-based temporal modules. Existing spatio-temporal graph learning frameworks combine graph convolution operations with temporal sequence modeling to capture coupled spatial interactions and temporal evolution in dynamic graph systems [61]. Recent airspace-specific studies further extend these architectures using adaptive graph convolution and temporal attention mechanisms for airspace complexity prediction and large-scale airspace modeling [50].

Mapping to UAV Constraints: These architectures are essential for applications requiring prediction of future states under mobility-induced topology changes and decentralized flocking behavior in multi-agent robotic systems [49,50,62,63].

Applications:

- Airspace traffic management: Model airspace complexity, trajectory evolution, and sector-level spatial interactions using aviation-focused spatio-temporal graph models [50].
- UAV dynamics: Temporal components capture acceleration and banking maneuvers.
- Neighbor coordination: Spatial graph convolution supports coordinated motion and decentralized flocking behavior in multi-agent systems [62,64].

Distinction from Dynamic GNNs

Although both Spatio-Temporal Graph Neural Networks and Dynamic Graph Neural Networks model evolving graph information, they address different types of graph dynamics. Table 3 summarizes the primary differences between these two architectures.

Table 3. Distinction between Spatio-Temporal GNNs and Dynamic GNNs.

Aspect	Spatio-Temporal GNNs	Dynamic GNNs
Node set	Fixed	Time-varying
Features	Time-varying	Time-varying
Edges	Time-varying	Time-varying with semantic evolution
Structural changes	Assumes fixed nodes	Explicit node additions/deletions
Best for	Fixed swarm with mobility	Highly mobile swarms with join/leave

Spatio-temporal GNNs assume a fixed node set with time-varying features and edges. Dynamic GNNs explicitly model structural changes, including node additions/deletions

and semantic edge evolution, and are particularly suitable for highly mobile UAV swarms where communication links form and break rapidly [65–70].

This distinction is important for UAV system design because the appropriate model class depends on the level of structural variability in the network. Spatio-temporal GNNs are more suitable when the UAV set remains fixed and only motion-dependent interactions evolve over time, whereas dynamic GNNs are better aligned with highly mobile swarm environments in which nodes may join, leave, or frequently lose connectivity. Thus, model selection should be informed by whether temporal evolution occurs primarily in node states or in the graph structure itself.

2.1.5. Distributed Graph Neural Networks

Distributed GNNs support decentralized computation across multiple UAVs without requiring centralized aggregation. Each UAV performs local message passing with neighbors and updates its own embeddings through federated learning or gossip-based protocols [36,71,72]. The layer-wise propagation is computed locally at each node [36], as defined in Equation (6):

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{W}^{(l)} \cdot \text{AGG}\left(\{\mathbf{h}_j^{(l)} : j \in \mathcal{N}(i)\}\right)\right) \quad (6)$$

where each UAV i computes aggregation using only neighbor states $\{\mathbf{h}_j\}$, avoiding global graph knowledge.

Mapping to UAV Constraints: Distributed architectures address critical UAV limitations. Federated GNN learning enables UAVs to train local models on subgraphs and share gradient updates rather than raw data, preserving privacy and reducing communication overhead in reported studies [71–73]. Over-the-air aggregation exploits wireless analog transmission for efficient gradient aggregation without digital protocols [34,74]. Distributed GNN systems such as DistDGL [36] demonstrate graph-partitioning strategies for scalable GNN training. Although not UAV-specific, these methods provide a transferable technical background for decentralized and large-scale graph learning, where large graphs are divided into subgraphs processed independently, with boundary nodes sharing representations [36]. In UAV systems, related distributed and over-the-air aggregation strategies provide relevant background for decentralized learning under communication constraints [37]. Decentralized architectures eliminate single points of failure, ensuring swarm robustness to individual UAV failures or communication disruptions [71].

Trade-offs include increased convergence time and potential accuracy degradation due to asynchronous updates and gradient staleness [71,75]. Applications include decentralized trajectory planning [34,38], privacy-preserving edge learning in UAV-MEC systems [20,76,77], and wireless gradient aggregation for resource-constrained swarms [72,74,78,79].

2.1.6. Resource-Efficient Graph Neural Networks

Resource-efficient GNNs reduce computational and memory requirements to enable deployment on embedded UAV platforms with limited processing power, memory, and battery life [19,20]. Model quantization reduces precision from 32-bit floating point to 8-bit integers, achieving 4–8× memory reduction with <3% accuracy loss [19,80]. Binary quantization achieves 32× compression but incurs 5–10% accuracy loss [81], while mixed-precision approaches balance efficiency and performance by using FP16 for critical layers and INT8 elsewhere [82].

Knowledge distillation trains compact student GNNs from large teacher models [83]:

$$\mathcal{L}_{\text{distill}} = \alpha \cdot \mathcal{L}_{\text{task}} + (1 - \alpha) \cdot \text{KL}(P_{\text{student}} \| P_{\text{teacher}}) \quad (7)$$

Knowledge distillation enables substantial parameter reduction while retaining most of the teacher model accuracy. Pruning removes redundant connections through structured or unstructured approaches, reducing computational complexity and improving efficiency. GNN-specific hardware accelerators such as AWB-GCN [84] and HyGCN [85] demonstrate substantial speedup and improved hardware efficiency for graph convolutional workloads. Although these accelerators are not evaluated specifically on UAV platforms, they provide relevant hardware-acceleration background for deploying resource-efficient GNN inference on constrained edge devices.

Mapping to UAV Constraints: Resource efficiency is critical for onboard deployment. Efficient GNN implementations enable real-time inference on embedded platforms such as Jetson TX2 and Raspberry Pi, though performance depends on model size and hardware optimization. As an order-of-magnitude illustration rather than a fixed rule, a mid-range embedded UAV computer such as the NVIDIA Jetson TX2 or NVIDIA Jetson Xavier NX provides on the order of 8 GB of shared memory, which in practice favors models in the tens-of-megabytes range once the operating system, perception stack, and runtime buffers are accounted for. The exact budget is platform-dependent: microcontroller-class autopilots provide far less, while larger companion computers provide more. Knowledge distillation can substantially shrink large teacher models to fit such budgets [83]. These figures should therefore be interpreted as platform-dependent guidance rather than a universal threshold. Energy-efficient inference extends flight time through reduced power consumption [86]. Lightweight GNNs can achieve high frame rates for real-time collision avoidance and navigation tasks [20]. Lightweight federated graph learning [20] demonstrates substantial memory reduction and faster inference with minimal accuracy loss for UAV-MEC systems. Applications include real-time collision avoidance, energy-constrained missions [86], and privacy-preserving edge learning [20,76].

Figure 4 presents a unified taxonomy of Graph Neural Network (GNN) architectures tailored for UAV applications, highlighting how different model families address distinct operational requirements and constraints. The taxonomy emphasizes that no single GNN architecture is universally optimal; instead, model selection depends on application demands, communication conditions, and onboard computational limitations. It should be noted that while GCN, GAT, GraphSAGE, and Spatio-Temporal GNNs represent distinct architectural families defined by their mathematical formulation and message-passing mechanisms, Distributed and Resource-Efficient GNNs represent deployment-oriented design paradigms applicable across multiple architectural families. They are included in this taxonomy due to their critical practical significance for UAV system deployment under real-world resource and communication constraints [19,20].

Convolutional GNNs, such as GCN and GraphSAGE, are primarily suited for scalable swarm coordination due to their reliance on localized neighborhood aggregation, which enables efficient processing and generalization across varying swarm sizes. However, their uniform treatment of neighbors limits their ability to capture heterogeneous interactions.

Attentional GNNs address this limitation by introducing adaptive weighting mechanisms, allowing UAVs to prioritize more informative or reliable neighbors. This makes them particularly effective in heterogeneous environments, although at the cost of increased computational complexity.

Sampling-based approaches, such as Cluster-GCN and GraphSAINT, improve scalability by enabling efficient subgraph sampling and mini-batch training, reducing the computational burden for large-scale swarm deployments.

Spatio-temporal and dynamic GNNs extend these capabilities by incorporating temporal evolution and topology changes, making them essential for applications such as

trajectory prediction and air traffic management. However, these models introduce additional latency and computational overhead.

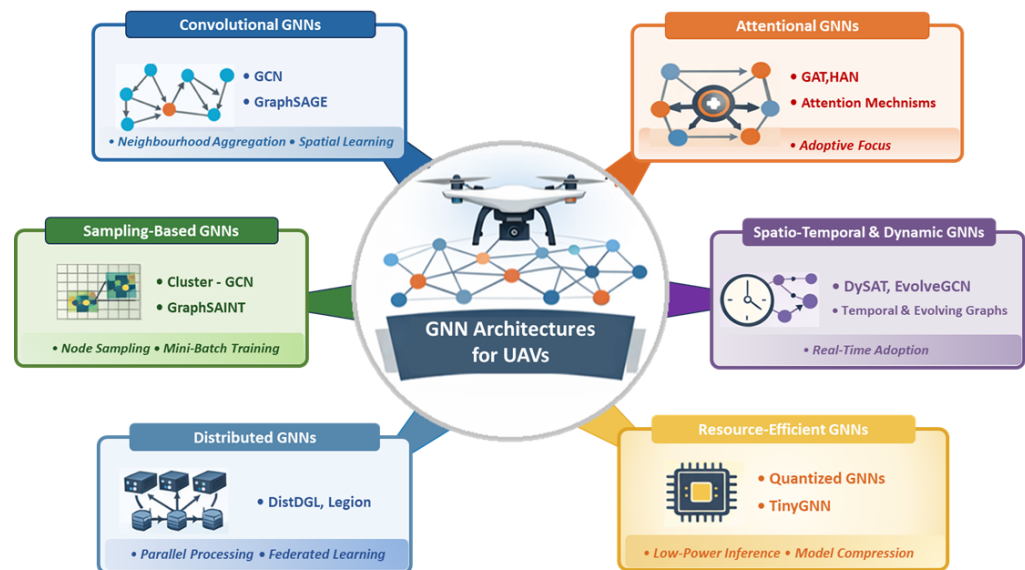


Figure 4. Architecture and Deployment Taxonomy of Graph Neural Networks for UAV Applications. Note: GCN, GAT, GraphSAGE, and ST-GNN represent architectural families defined by their mathematical formulation, while Distributed and Resource-Efficient GNNs represent deployment-oriented paradigms applicable across multiple architectures.

Finally, distributed and resource-efficient GNN frameworks address practical deployment constraints by enabling decentralized learning and reducing model complexity for embedded UAV platforms. Overall, the taxonomy highlights that GNN architecture selection involves a fundamental trade-off between scalability, adaptability, and computational efficiency, requiring careful alignment with application-specific constraints.

2.1.7. Emerging Architectures

Three further architecture classes are receiving increasing attention and are relevant to UAV systems, although their UAV-specific evidence base remains limited. Graph Transformers replace local message passing with global self-attention over nodes, which mitigates over-smoothing and the limited receptive field of deep convolutional GNNs (Section 4.1). For UAV swarms, this becomes important when coordination requires dependencies spanning more hops than a shallow GCN can capture, such as global formation consensus. The tradeoff is that attention typically scales quadratically with the number of nodes, which conflicts with onboard computational constraints and currently restricts practical deployment to small swarms or sparsified-attention variants [87,88].

Graph diffusion models define propagation or generation through a diffusion process over the graph. As a propagation mechanism, they improve long-range information flow [89]; as a generative mechanism, they can produce candidate trajectories or formations under constraints [90]. Their relevance to UAV trajectory generation is promising, although evidence in aerial systems remains limited.

Multi-modal graph learning integrates heterogeneous inputs such as vision, range sensing, and communication state within a unified graph representation, closely matching the multi-sensor nature of UAV platforms [91]. The main challenge remains performing such fusion within strict onboard compute and energy budgets.

These architecture classes are not given dedicated application subsections because UAV-specific empirical evidence remains sparse. They are included here to identify emerg-

ing research directions rather than established practice, consistent with the inclusion criteria discussed in Section 1.5.

2.2. Formal UAV System Models and Problem Definitions

2.2.1. Mathematical Problem Formulations

This section formalizes three canonical UAV system problems and demonstrates their natural correspondence to graph-based learning formulations.

Problem 1: Multi-UAV Trajectory Planning

Consider a set of UAVs $\mathcal{V} = \{1, 2, \dots, N\}$, where each UAV i has state $\mathbf{x}_i(t)$ and control input $\mathbf{u}_i(t)$. The objective is to generate collision-free trajectories from initial states to predefined goal locations while maintaining communication connectivity.

The dynamic interaction among UAVs is modeled as a time-varying graph:

$$\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t) \quad (8)$$

where edges represent communication or proximity relationships [18,24].

The trajectory optimization problem is formulated as:

$$\min_{\mathbf{u}_i(t)} \sum_{i=1}^N \int_0^T c(\mathbf{x}_i(t), \mathbf{u}_i(t)) dt \quad (9)$$

subject to

- Collision avoidance constraints:

$$\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| \geq d_{\text{safe}}, \quad \forall i \neq j, \quad (10)$$

- UAV dynamics:

$$\dot{\mathbf{x}}_i(t) = f(\mathbf{x}_i(t), \mathbf{u}_i(t)), \quad (11)$$

- Connectivity preservation induced by $\mathcal{G}_t$.

Trajectory planning and collision avoidance are extensively studied in UAV systems [92–94], while recent work integrates graph-based learning for decentralized coordination [27,38,41,55].

Problem 2: UAV Communication Network Optimization

A UAV communication network is modeled as a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where edge weights correspond to channel gains h_{ij} .

The signal-to-interference-plus-noise ratio (SINR) for UAV i is defined as:

$$\text{SINR}_i = \frac{P_i h_{ii}}{\sum_{j \neq i} P_j h_{ji} + \sigma^2} \quad (12)$$

where P_i denotes transmission power and σ^2 represents noise power.

The objective is to maximize network throughput:

$$\max_{\{P_i\}_{i=1}^N} \sum_{i=1}^N \log(1 + \text{SINR}_i) \quad (13)$$

This formulation is widely adopted in UAV-enabled wireless networks [1,31,32,95].

Graph neural networks have recently been applied to such problems by learning mappings from interference graphs to resource allocation policies [10,11,15,16,96,97].

Problem 3: Cooperative Multi-Target Tracking

Let $\mathcal{U}$ denote the set of UAVs and $\mathcal{T}$ the set of targets. The sensing relationship is represented as a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \mathcal{U} \cup \mathcal{T}$ and edges indicate observable UAV–target pairs.

The objective is to minimize estimation uncertainty:

$$\min \text{Tr}(\mathbf{P}) \quad (14)$$

where $\mathbf{P}$ is the covariance matrix of the estimated target states.

Multi-target tracking in UAV systems has been extensively studied using control and reinforcement learning frameworks [43,45,98,99]. Recent approaches incorporate graph neural networks to model inter-agent and agent-target dependencies, improving tracking robustness and coordination [42,44,100,101].

2.2.2. UAV System Models

Table 4 summarizes UAV subsystem characteristics and their graph-theoretic representations.

Table 4. UAV System Characteristics and Graph Representations.

UAV Subsystem	Graph Type	Nodes	Edges	Features	Learning Objective	References
Communication Network	Dynamic, Weighted	UAVs	Connectivity links	SINR, throughput, delay	Minimize latency, maximize capacity	[15,16,26,46,47]
Navigation & Control	Static/Temporal	UAVs, Waypoints	Proximity, precedence	Position, velocity, heading	Collision-free path, energy-optimal	[39–41,92,93]
Swarm Coordination	Dynamic	UAVs	Communication range	State, task assignment	Formation stability, consensus	[27,28,53,102,103]
Target Tracking	Bipartite, Temporal	UAVs, Targets	Observation links	Sensor measurements, beliefs	Minimize tracking error	[42–45,98]
Resource Allocation	Static, Weighted	UAVs, Tasks	Capability match	Compute, energy, payload	Maximize task completion	[104–108]

Communication Model: UAV-to-UAV links follow free-space path loss, where d is distance and λ is wavelength [95,109]. Link quality varies with mobility, requiring continuous graph adaptation [18,24].

Navigation Model: UAV dynamics follow motion constraints with control bounded by system limits [92]. Safe distance enforces collision avoidance [94].

Swarm Coordination: Inspired by Reynolds’ flocking model [53], UAVs maintain graph-based coordination objectives [8,27,110].

Table 4 highlights that different UAV subsystems naturally correspond to different graph formulations, feature spaces, and learning objectives. Communication and resource allocation problems are more naturally modeled using weighted graphs that encode interference, capacity, or capability relationships, whereas navigation and swarm coordination rely more strongly on spatial or temporal interaction graphs. This mapping suggests that effective GNN design should begin with a careful abstraction of the physical UAV subsystem into an appropriate graph representation, rather than applying a generic architecture across all tasks.

To complement this graph-based abstraction, Figure 5 illustrates a system-level architecture of GNN-enabled UAV networks, showing how local sensing, graph construction, lightweight onboard inference, peer-to-peer communication, and optional centralized or edge-based training interact under practical deployment constraints.

As shown in Figure 5, GNN-enabled UAV systems can be viewed as dynamic communication graphs in which each UAV performs local sensing, graph construction, and lightweight inference while adapting to communication variability and resource constraints.

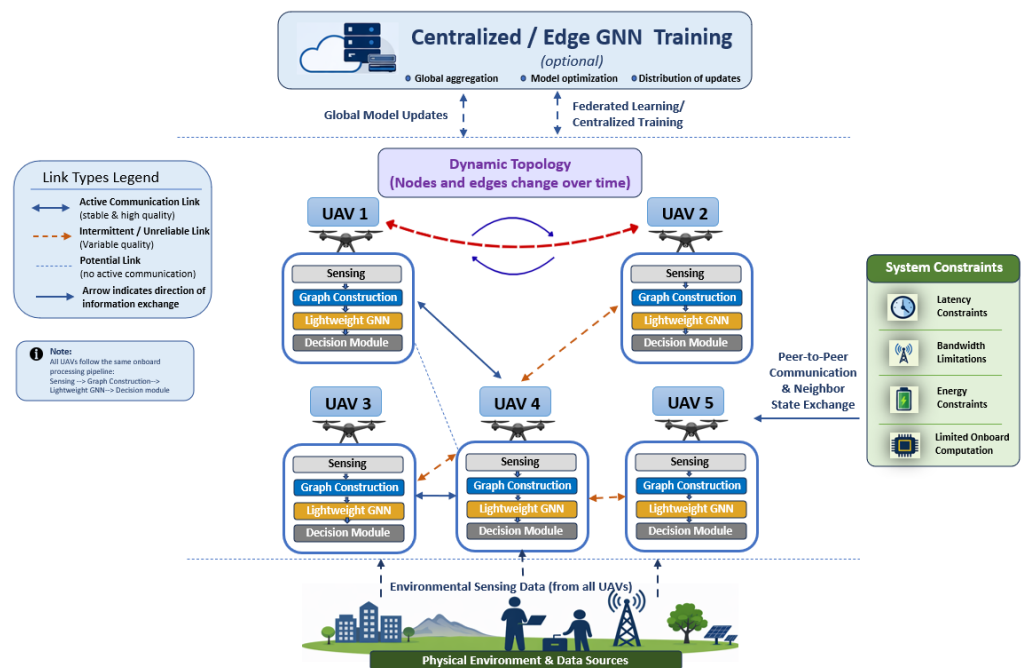


Figure 5. GNN-enabled UAV system architecture. Multiple UAVs operate as nodes in a dynamic communication graph, where edges represent active, intermittent, or potential communication links. Each UAV performs onboard sensing, graph construction, lightweight GNN inference, and local decision-making. Peer-to-peer message passing supports decentralized coordination, while optional centralized or edge-based training enables global model aggregation and distribution of updates. The architecture operates under practical constraints including latency, bandwidth, energy, and limited onboard computation.

2.3. Resource-Constrained GNN Design

Practical deployment demands resource-efficient architectures due to strict onboard energy, memory, and computational constraints [19]. Model quantization reduces precision to 8-bit, decreasing memory by approximately 75% [80–82,111]. Knowledge distillation transfers knowledge to compact networks for embedded processors [83]. Hardware acceleration achieves real-time inference within UAV power budgets [84–86]. These optimizations extend flight time and enable real-time decisions [112–114]. Distributed over-the-air training enables collaborative learning without centralizing data [34,36,37,74].

3. Applications: GNN-Enabled Intelligence for UAV Systems

Following the establishment of mathematical underpinnings and architectural design guidelines in Section 2, GNN deployment is analyzed across several UAV application domains. This section examines how graph-based solutions solve real-world problems in multi-UAV systems by breaking them down into four application areas: trajectory planning and collision avoidance, cooperative target monitoring and surveillance, communication network optimization, and spatio-temporal air traffic prediction.

3.1. Multi-UAV Trajectory Planning and Collision Avoidance

Multi-UAV trajectory planning maintains agents in three-dimensional space while ensuring collision avoidance, communication connectivity, kinodynamic feasibility, and mission completion. Traditional centralized planning is computationally intractable as swarm size increases, whereas purely reactive approaches often produce suboptimal trajectories. Graph neural networks facilitate distributed yet coordinated planning through learned graph-structured policies [38,41].

3.1.1. Graph-Based Planning Frameworks

Learning-Based Cooperative Planning: Frameworks combining GNNs with reinforcement learning process relative positions and interactions through communication graphs [41]. GNN models trained on small swarms generalize to larger formations without retraining, directly addressing scalability challenges [27,28].

Attention-Driven Coordination: Graph attention mechanisms dynamically weight neighbor importance, enabling adaptive coordination in heterogeneous swarms [39]. In environments with limited communication, attention weights prioritize high-quality information sources.

Hybrid Approaches: Several studies integrate GNNs with classical optimization frameworks [55,115]. In these approaches, GNNs provide learned priors to guide optimization solvers, reducing computational complexity while preserving feasibility constraints.

Domain-Specific Adaptation: Advanced frameworks incorporate environmental context into graph structures [40,116–118]. For infrastructure inspection tasks [116], graph connectivity may depend on visibility constraints, while confined environments require geometry-aware connectivity modeling [117].

GNN-based MARL for Coordination and Task Allocation: Recent advances in graph neural network (GNN)-based multi-agent reinforcement learning (MARL) have significantly improved coordination and decision-making in multi-UAV systems. For instance, the MAGNNET framework integrates GNNs with a centralized training and decentralized execution (CTDE) paradigm, enabling agents to learn globally optimal policies during training while making decisions independently during deployment [119]. This approach has demonstrated strong performance in decentralized task allocation problems, achieving high conflict-free task assignment rates and maintaining scalability in dynamic environments.

In addition to task allocation, GNN-based MARL approaches have been applied to resilient multi-agent coordination under real-world uncertainties. Recent work demonstrates that graph-based coordination mechanisms remain robust under agent failures, communication disturbances, and partial observability [35]. Such approaches highlight the importance of designing coordination strategies that are not only efficient but also resilient to real-world operational challenges.

3.1.2. Collision Avoidance Strategies

Reactive Neural Avoidance: Sensor-to-control pipelines augmented with graph-based coordination ensure that individual avoidance actions remain compatible with swarm-level objectives [29].

Predictive Planning: Temporal prediction models estimate future UAV positions, enabling proactive maneuver planning and smoother collision avoidance [120].

Learning-Enhanced Methods: Several works combine classical Velocity Obstacle methods [94] with learning-based components [30], achieving improved safety–efficiency trade-offs.

Multi-Objective Coordination: Modern systems optimize multiple objectives simultaneously, including flight time, energy consumption, and formation stability [112,121,122].

3.1.3. Localization and SLAM

Localization and SLAM: Graph-based representations are fundamental to simultaneous localization and mapping (SLAM), where UAV states and environmental landmarks are naturally modeled as nodes in a graph. Emerging approaches explore the integration of graph neural networks with SLAM pipelines to improve state estimation, loop closure detection, and robustness in dynamic and partially observable environments. While SLAM focuses on environment representation and localization, trajectory-planning methods pri-

marily address coordination and control strategies in multi-UAV systems, as summarized in Table 5.

Table 5. Categorization of GNN-Based Multi-UAV Trajectory-Planning Methods.

Primary Method	GNN Architecture	Application Domain	Key Innovation	References
Reinforcement Learning	GCN/GraphSAGE + RL	General coordination	Scalable policy learning and transfer to larger swarms	[38–41]
Integration	GCN + MPC/A* search	Constrained planning	Combines learning efficiency with formal guarantees	[55,115]
Hybrid	Temporal GNN	Dynamic environments	Multi-step ahead prediction for proactive planning	[120,123]
Learning-Optimization	GAT + RL	Heterogeneous swarms	Dynamic neighbor importance weighting	[39]
Temporal Prediction	Various	Inspection, delivery, confined spaces	Context-aware graph construction	[116–118,121,122]
Attention Mechanisms	GNN + Vision	Sensor-based navigation	Direct sensor-to-action with multi-agent coordination	[29,30]
Domain-Specific Adaptation				
Reactive Coordination				

Table 5 reveals clear trade-offs among GNN-based trajectory-planning approaches. Learning-based methods provide strong scalability and fast inference, but they often lack explicit safety guarantees, which limits their suitability for safety-critical missions. Hybrid learning-optimization approaches offer a more balanced solution by combining learned efficiency with constraint enforcement, making them particularly attractive for real-world UAV deployment. Reinforcement learning-based methods provide adaptability in dynamic environments, but their training instability and computational cost remain significant barriers. Therefore, method selection should be guided by the relative importance of safety, scalability, computational efficiency, and environmental complexity.

3.1.4. Efficiency of Computation

GNN-based methodologies achieve favorable scalability through localized computation [28]. Each UAV independently processes a small neighborhood of nearby agents (typically five to ten neighbors), regardless of the overall swarm size. Hardware acceleration further enables real-time deployment on resource-constrained platforms [19].

Planning and tracking are not independent: the proximity graphs used for collision-free planning are the same structures that, once targets are added as nodes, support the cooperative tracking discussed next.

3.2. Collaborative Target Monitoring and Surveillance

Multi-UAV tracking leverages spatial diversity to improve localization accuracy and maintain persistent surveillance coverage [42–45].

3.2.1. Multi-Target Tracking

Heterogeneous Graphs: Tracking frameworks often model heterogeneous graphs containing multiple node types (UAVs, targets, and observations) and edge types (observation, communication, and temporal relationships), enabling integrated data association and estimation [42].

Cross-Attention Fusion: Attention-based mechanisms adaptively weight sensor measurements according to sensing quality and spatial geometry [44,100].

Vision–GNN Integration: Recent work has explored integrating visual perception with graph neural networks, where object-level features extracted from onboard cameras are represented as scene graphs for decision-making. Such vision–GNN pipelines enable obstacle-aware navigation and improve robustness in complex and dynamic environments.

Belief Propagation: Graph neural networks can perform approximate belief propagation to jointly solve estimation and coordination problems in distributed tracking systems [43,45].

3.2.2. Search and Surveillance

MARL for Search: Graph-based multi-agent reinforcement learning (MARL) approaches learn coordinated search policies that balance exploration and coverage [101,124]. Local GNN interaction rules can produce emergent behaviors such as region partitioning and cooperative target handoffs [124].

Long-Duration Monitoring: Temporal and graph-based methods can support persistent monitoring by modeling evolving relationships among UAVs, targets, and observed regions over time. UAV-based monitoring applications have been widely studied in environmental sensing and long-duration missions [125]. In addition, UAV swarm communication studies highlight the importance of UAV-to-UAV communication, FANET coordination, distributed decision-making, and reliable communication architectures for persistent multi-UAV operation [102]. However, these works focus on monitoring applications and swarm communication/control architectures rather than dynamic GNN-based monitoring methods. Therefore, dynamic GNN-based long-duration monitoring remains an open research direction.

UAV Detection and Classification: Recent research has explored graph-based learning for UAV detection and classification using external sensing modalities such as radar, radio frequency (RF), and acoustic signals. By modeling sensor observations and inter-sensor relationships as graphs, GNN-based approaches enable robust detection, classification, and tracking of UAVs in complex and noisy environments.

3.3. Communication-Aware Graph Modeling

UAV communication networks must handle dynamically changing topologies, time-varying channels, interference management, and strict latency requirements [15–17].

GNN and MARL for UAV Communication Networks: In UAV-assisted communication networks, recent research has explored the integration of multi-agent reinforcement learning (MARL) and graph-based learning to address large-scale coordination challenges. A notable direction involves combining MARL with knowledge distillation from large language models (LLMs) to enhance decision-making efficiency and scalability in multi-UAV networking scenarios [126]. These hybrid frameworks leverage high-level guidance during training while maintaining decentralized execution, resulting in improved communication coverage, network robustness, and adaptability in highly dynamic environments.

FANET Optimization: GNN-based approaches jointly optimize UAV trajectories and network topology to maintain connectivity in flying ad hoc networks (FANETs) [26,46]. Such systems can detect performance degradation and autonomously trigger network reconfiguration [47]. Learned GNN policies also support predictive caching and resource provisioning in UAV-assisted networks [48].

Resource Allocation: Graph-based models represent wireless interference relationships to optimize power allocation and spectrum usage [96,97]. Knowledge graph completion techniques can infer missing channel information [127]. GNN-based frameworks also balance computation and communication costs during task offloading in edge-computing environments [128].

Routing and Network Design: GNN-enabled routing algorithms determine optimal forwarding strategies based on network topology and traffic patterns [129]. Enhanced models incorporate link-quality metrics in addition to geometric connectivity [21]. Cross-layer frameworks jointly optimize decisions across physical, MAC, and network layers [22].

Bipartite graph formulations have also been applied to optimal UAV placement for communication coverage [130,131].

The communication graphs discussed above are largely static snapshots. When the evolution of these graphs over time becomes the primary quantity of interest, the spatio-temporal models discussed in the next subsection become more appropriate.

3.4. Spatio-Temporal GNNs for Air Traffic

As UAV traffic density increases, reliable traffic prediction and conflict detection become critical challenges [49,50,132,133].

Trajectory Prediction: Spatio-temporal graph neural networks (ST-GNNs) capture spatial dependencies and temporal dynamics in trajectory data [132]. Multi-modal systems combine radar observations, flight plans, and meteorological data to improve prediction accuracy [50]. Airspace complexity metrics account for both velocity dynamics and interaction patterns among aircraft [50].

Large-Scale Predictions: Spatio-temporal models enable large-scale prediction of traffic flow and density patterns [64,133]. Federated learning frameworks allow collaborative model training across distributed systems while preserving data privacy [49]. Hierarchical architectures capture multi-resolution temporal dynamics. Similar techniques have also been applied to terrestrial transportation networks and intelligent urban environments [134].

However, it is important to note that many existing spatio-temporal GNN models are validated on terrestrial transportation datasets, and there remains a lack of standardized large-scale UAV air traffic datasets for direct evaluation.

It should be noted that the traffic prediction studies summarized here are based on diverse datasets and evaluation settings, and therefore provide qualitative guidance rather than direct quantitative comparison across UAV-specific benchmarks.

Unlike a single empirical comparison, Table 6 synthesizes trends across multiple studies to provide evidence-informed guidelines for model selection. It highlights that architecture choice is inherently application-dependent, requiring alignment with system scale, communication dynamics, and computational constraints rather than relying on a universally optimal model.

Table 6. Synthesized GNN Architecture-Selection Guidelines by Application Domain. Individual recommendations originate from cited primary sources; the contribution lies in unified cross-domain organization enabling systematic architecture comparison.

Application Domain	Primary Challenge	Recommended GNN Families	Architectural Rationale	Key References
Trajectory Planning	Scalable coordination	GCN, GraphSAGE + RL	Permutation equivariance supports consistent node-level policies across varying swarm sizes	[28,38–41]
Target Tracking	Multi-sensor fusion	GAT, Heterogeneous GNN	Attention weights observations by quality; heterogeneous structure represents UAV–target relationships	[42–45,100]
FANET Routing	Topology adaptation	Dynamic GNN, GCN	Handles rapid connectivity changes and learns routing policies from traffic patterns	[21–23,47,129]
Spectrum Management	Interference mitigation	GCN, Graph RL	Interference graph structure directly informs power/channel optimization	[96,97,127,128]

Table 6. *Cont.*

Application Domain	Primary Challenge	Recommended GNN Families	Architectural Rationale	Key References
Swarm Coordination	Distributed consensus	GCN, Dynamic GNN	Local computation with message passing enables distributed execution	[27,28,53,102,103]
Traffic Prediction	Spatio-temporal forecasting	ST-GNN, Temporal GNN	Joint spatial-temporal modeling captures coordination dynamics and motion patterns	[49,50,132,133]
Task Allocation	Combinatorial optimization	GNN + RL, GraphSAGE	Learns assignment heuristics; inductive learning handles variable task/UAV counts	[104–108]
Search & Surveillance	Exploration-exploitation	MARL + GNN	Multi-agent learning discovers coordinated search strategies	[98,101,124,125]

3.5. Patterns of Research and Cross-Cutting Themes

The application landscape reveals several recurring themes that extend beyond individual domains. These themes influence both research directions and practical deployment strategies in multi-UAV systems [14,27,28,56].

Architecture-Application Matching: Different GNN architectures are better suited to particular UAV tasks. Graph attention networks are effective when dynamic neighbor weighting is required [58,135]. This is especially relevant in target tracking scenarios with heterogeneous sensor quality [42–45]. It is also useful in heterogeneous UAV swarms with varying capabilities [28,136]. When spatial coordination and temporal dynamics must be modeled jointly, spatio-temporal architectures become important for UAV trajectory prediction, airspace monitoring, and coordinated motion analysis [49,50,63]. Recent airspace complexity prediction studies further demonstrate that multimodal spatio-temporal graph neural networks can effectively capture coupled spatial interactions and temporal evolution in large-scale airspace systems [50]. These models are particularly useful for airspace traffic prediction, trajectory forecasting, and airspace management [50,64,132,133]. Inductive learning approaches such as GraphSAGE enable generalization across varying swarm sizes [33,60]. This capability is important for scalable swarm coordination and dynamic task allocation [27,28,104–107].

Read together, these studies trace a single research trajectory rather than a collection of isolated results. Early work established that graph structure improves coordination relative to non-relational baselines; later work shifted the question from whether GNNs help to which architecture performs best under specific constraints, with attention-based models excelling when link quality varies and sampling-based models performing better when swarm size changes. The remaining disagreement in the literature is therefore not about the value of graph structure, but about which inductive bias best matches a given operating condition.

Hybrid Learning-Optimization Paradigms: Practical implementations increasingly adopt hybrid frameworks that integrate learned models with classical optimization strategies [41,137–140]. Pure learning approaches offer flexibility and computational efficiency [14,56]. However, they may struggle to enforce strict safety constraints [29,38,40]. Conversely, purely optimization-based methods can become computationally prohibitive as graph size increases. Large-scale GNN systems provide transferable background on graph partitioning, caching, and scalable training, although they are not direct UAV deployment evidence [36,141,142]. Hybrid approaches use GNNs to provide learned priors, heuristics, or warm-start solutions for optimization solvers [35,41,138–140]. This combination improves both efficiency and reliability, which is important for safety-critical aerial operations [29,38–40,55].

Distributed Implementation Patterns: A common implementation paradigm relies on distributed computation. In this setting, each UAV processes only locally available information through GNN message passing [27,34,36,47,56]. This approach aligns naturally with coordination under limited communication conditions [15,16,28,47]. It also supports onboard computational constraints [19,20,80,111]. Because each UAV performs localized computation, the per-agent computational cost remains approximately constant as swarm size increases. This property supports more efficient multi-UAV coordination in UAV-specific studies [27,28]. General large-scale GNN systems further provide transferable technical background for distributed graph processing and training scalability [36,141,142].

Integration of Multi-Source Data: Advanced UAV systems often integrate heterogeneous data sources. These include optical and LiDAR observations [43–45], communication metrics such as SINR and latency [15,16,46,47], and mission-level variables such as task assignments and energy levels [104–107]. Heterogeneous graph representations support diverse node types, including UAVs, targets, tasks, and ground stations [58,136]. They also represent multiple edge types, such as communication links, observations, and task dependencies. This structure enables unified reasoning across complex multi-modal UAV environments [43–45,58,136].

Transfer Learning and Generalization: A key advantage of GNN-based methods is their ability to generalize across different swarm sizes, environmental conditions, and mission types [27,28,33,60,104–106]. In some cases, learned representations can also transfer across application domains [14,56]. For example, trajectory-planning models may inform tracking applications [42–45]. Similarly, communication-optimization methods can be adapted for routing or coverage problems [15,16,46,47,131]. Such transfer can reduce development time and improve system robustness [33,60].

Scalability via Graph Structure: The inherent sparsity of graph-based representations provides strong scalability benefits [14,56]. Unlike fully connected neural networks, GNNs operate on sparse interaction graphs. In these graphs, nodes communicate primarily with local neighbors [14,27,28,47]. This sparsity reduces computational complexity and supports distributed implementation [34,36,142]. As a result, UAV-specific studies suggest improved scalability for multi-agent coordination [27,28].

Scalability mechanisms borrowed from general-purpose GNN systems do not transfer directly to UAV environments; Section 5.1 analyzes these discrepancies and their bottlenecks in detail.

Overall, these patterns highlight that graph neural networks provide a unified modeling framework for multi-UAV systems. They enable decentralized coordination, efficient representation of local interactions, and integration of heterogeneous data sources. These properties make GNNs particularly suitable for large-scale swarm systems where centralized control becomes impractical.

Taken together, these cross-cutting patterns show that GNN-based UAV research is moving from isolated task-specific models toward more integrated design principles. Section 3.6 builds on these observations by comparing the strengths and weaknesses of major GNN families across UAV system requirements.

3.6. Comparative Analysis of GNN Approaches in UAV Systems

Building on the cross-cutting patterns identified in Section 3.5, this subsection compares major GNN architecture families from a model-selection perspective, focusing on scalability, adaptability, computational feasibility, and real-world applicability.

Convolutional GNNs (GCN, GraphSAGE): Convolution-based models such as GCN and GraphSAGE are widely adopted due to their scalability and efficient localized message passing [14,33]. These architectures support decentralized coordination and enable

generalization across varying swarm sizes, making them suitable for large-scale UAV systems [27,28]. This design emphasizes efficiency and robustness, though it is less expressive in highly heterogeneous environments where node importance varies dynamically.

Attentional GNNs (GAT, HAN): Attention-based models introduce adaptive weighting mechanisms that allow UAVs to prioritize more informative neighbors [57,58]. This improves performance in scenarios with heterogeneous sensing quality or asymmetric communication. These models offer enhanced flexibility, although they typically incur higher computational cost compared to convolutional approaches [135].

Sampling-Based GNNs: Sampling-based methods, such as Cluster-GCN and adaptive neighborhood sampling, improve scalability by reducing the computational burden of message passing [56,143,144]. These approaches are particularly effective for large-scale graphs, although sampling strategies may introduce approximation effects depending on the application context.

Spatio-Temporal and Dynamic GNNs: Spatio-temporal and dynamic GNNs extend graph learning to time-varying systems by modeling evolving node interactions and temporal dependencies [67,68,70]. These architectures are well suited for trajectory prediction, airspace monitoring, and adaptive coordination tasks [50,132]. Their effectiveness depends on the availability of consistent temporal data and stable communication patterns.

Distributed and Federated GNNs: Distributed and federated learning frameworks align naturally with decentralized UAV systems by enabling collaborative model training without centralized data aggregation [20,71,73]. These approaches improve scalability and data privacy while introducing additional considerations related to communication efficiency and synchronization [34].

Resource-Efficient GNNs: Resource-efficient GNNs focus on lightweight architectures, quantization, and model compression techniques to enable deployment on embedded UAV platforms [19,111]. These methods improve real-time feasibility, though they require careful balancing between computational efficiency and model expressiveness.

Key Insight: No single GNN architecture is universally optimal for UAV systems. Convolutional models provide scalability, attention-based models improve adaptability, and dynamic architectures capture temporal evolution. Resource-efficient variants enhance deployment feasibility on constrained hardware platforms.

To complement the application-level comparison above, Table 7 summarizes the computational and deployment tradeoffs across major GNN architecture families, focusing on asymptotic computational complexity, scalability, communication overhead, and onboard resource requirements relevant to UAV systems.

Table 7. Computational cost and deployment profile of major GNN architecture families for UAV systems.

Architecture Family	Per-Layer Computational Cost	Scalability to Large Swarms	Comm. Overhead (Decentralized)	Onboard Memory/Energy
GCN	$O(E F + V F^2)$	Moderate (full-graph, transductive)	Low (1-hop exchange)	Low–Moderate
GraphSAGE	$O(V \cdot S \cdot F + V F^2)$	High (inductive; sampling bounds cost)	Low	Low
GAT	$O(V FF' + E F')$	Moderate (grows with edge density)	Moderate (extra cost mainly compute)	Moderate
Spatio-Temporal	$O(T(E F + V F^2))$ + temporal module	Moderate (temporal window multiplies cost)	Moderate	High
Distributed	Partition-dependent + synchronization cost	High (designed for scale)	High (gradient/embedding sync)	Moderate
Resource-Efficient	Reduced $\approx 4\times$ at 8-bit	High	Low	Very Low

Note: F = feature dimension, F' = output dimension, S = neighbors sampled per layer, and T = time steps. Costs are asymptotic per-layer profiles synthesized from cited architecture papers (GCN: [14]; GAT: [57], which reports GAT complexity is comparable to GCN), rather than measured benchmark runtimes.

The selection of an appropriate model therefore depends on task requirements, communication conditions, and computational constraints. These comparative insights provide a foundation for understanding broader system-level challenges, which are discussed in Section 5.

3.7. Quantitative and Qualitative Performance Summary

To provide a structured perspective on the effectiveness of graph neural network (GNN)-based approaches in UAV and multi-agent systems, Table 8 presents a consolidated performance summary of representative studies published between 2018 and 2025. Only studies that explicitly employ GNN-based or graph-learning architectures in UAV or closely related multi-agent robotic contexts are included. Non-GNN works, general surveys, and purely road-traffic prediction benchmarks are excluded to ensure domain relevance and avoid misleading comparisons.

The studies included in Table 8 are selected as representative examples that report explicit quantitative performance metrics across different UAV application domains. The reported results illustrate performance across diverse applications, including trajectory planning, target tracking, communication optimization, task allocation, visual tracking, and swarm coordination.

However, datasets, simulation environments, evaluation protocols, and performance metrics vary significantly across studies. As a result, the values presented in Table 8 should be interpreted as a descriptive synthesis rather than a standardized benchmark. Direct numerical ranking of methods may be misleading because metrics such as success rate, collision rate, HOTA, MOTA, FPS, throughput improvement, and connectivity score measure different task objectives.

Despite these differences, several trends can be observed. GNN-based methods integrated with reinforcement learning demonstrate strong performance in decentralized coordination tasks, particularly in navigation, task allocation, and communication scheduling. Attention-based architectures show competitive results in perception-driven applications such as multi-object tracking and visual tracking, while spatio-temporal and hybrid models are particularly effective in dynamic environments requiring predictive capabilities.

At the same time, a notable portion of the literature reports qualitative or relative improvements without standardized numerical metrics. This reflects the early-stage nature of benchmarking in GNN-enabled UAV systems and highlights the need for unified evaluation protocols. Furthermore, most results are obtained in simulation environments, indicating that large-scale real-world validation remains an open challenge.

Overall, the available evidence supports the effectiveness of GNN-based approaches for multi-UAV intelligence while also emphasizing the importance of standardized benchmarks and deployment-oriented evaluation in future research.

Table 8 present representative studies selected from the 196 included works using the following criteria: (i) the study applies a GNN-based method to a UAV or closely related multi-agent task; (ii) its application falls within one of the four surveyed domains; and (iii) collectively the selected studies span all four domains and the main GNN families. Studies that report at least one numerical performance value are summarized quantitatively in Table 8.

Table 9 reframes the available evidence using relative change over a defined baseline, which provides a more comparable view than raw absolute values across heterogeneous metrics. Two patterns appear. First, the magnitude of improvement depends strongly on the baseline. Yang et al. [41] report a 44.0% gain in success rate over I-PPO, but a smaller 15.4% gain over the stronger MADDPG baseline in Scenario A. In the high-density scenario,

the same study reports a larger relative improvement over representative MARL baselines, with a 78.0% success-rate improvement and a 73% collision-rate reduction.

Table 8. Quantitative performance summary of GNN-based UAV and multi-agent systems (2018–2025).

Reference	Year	Application Area	Dataset	GNN Model	Task	Metric	Reported Performance
Yang et al. [41]	2025	Trajectory planning	Simulation (obstacle environment)	GCN + PPO	Multi-UAV navigation	Success rate/Collision rate	94.2% success; 5.1% collision
Yang et al. [115]	2025	Navigation	OmniDrones (simulation)	Graph RL	Scalable coordination	Task efficiency	27.9% improvement over baseline
Ji et al. [123]	2025	Path planning	Simulation (dynamic UAV environment)	GNN + U-Net	Dynamic trajectory generation	Success rate	92.8% success
Qian et al. [44]	2025	Target tracking	VisDrone, UAVDT	GAT-based model	Multi-object tracking	HOTA/MOTA	51.34% HOTA; 50.47% MOTA
Xue et al. [100]	2023	Visual tracking	Real-world + benchmark datasets	GNN-enhanced model	Small-object tracking	FPS	72.5 FPS
Hu et al. [145]	2025	Decision-making	Simulation (multi-agent environment)	GNN + MARL	UAV confrontation	Effectiveness	97.4% effectiveness
Ren et al. [146]	2023	Communication scheduling	Simulation (network topology)	Graph RL	Data transmission scheduling	Throughput	11% improvement over baseline
Mou et al. [47]	2022	Communication resilience	Simulation (network graph)	GCN	Connectivity maintenance	Connectivity (J_c)	0.311 vs. 0.077 baseline

Note: Reported performance values are task-specific and derived from different datasets, simulation settings, baselines, and evaluation metrics. Therefore, this table provides a descriptive synthesis rather than a standardized benchmark or cross-domain ranking.

Table 9. Relative performance changes reported or computable from representative GNN-based UAV studies.

Study	Domain	GNN Method	Baseline	Relative Performance Change
Yang et al. [41]	Trajectory planning, Scenario A	GNN-PPO	I-PPO	+44.0% success rate, computed from 94.2% vs. 65.4%
Yang et al. [41]	Trajectory planning, Scenario A	GNN-PPO	MADDPG	+15.4% success rate, computed from 94.2% vs. 81.6%
Yang et al. [41]	Trajectory planning, high-density Scenario B	GNN-PPO	Representative MARL baseline	+78.0% success-rate improvement, computed from 85.8% vs. 48.2%; 73% collision-rate reduction as reported
Yang et al. [115]	Multi-UAV navigation, OmniDrones-20	MASP, graph-based HRL	(H)Voronoi	27.9% fewer steps, computed from 315.31 vs. 437.43
Yang et al. [115]	Multi-UAV navigation, MPE-50	MASP, graph-based HRL	Planning-based baselines	18.0–19.1% fewer steps, computed from 47.46 vs. 57.88 ((H)Voronoi) and 58.68 ((H)RRT*))
Ren et al. [146]	Ground-to-air heterogeneous data communication	AGUZero, attention-based graph RL	Reported baseline models	+11.03% and +10.35% transmission-ratio improvement in two reported cases
Mou et al. [47]	Communication resilience under general UEDs	CR-MGCM, GCN-based	CEN, strongest non-proposed baseline reported in Figure 16 of Mou et al. [47]	+55.5% connected time-step ratio J_c , computed from 0.311 vs. 0.200

Note: Relative performance changes are computed from reported values in the cited studies whenever a defined baseline is available. Because datasets, environments, metrics, and baseline choices differ across studies, these comparisons provide indicative evidence rather than a standardized benchmark. RRT* denotes the Rapidly-exploring Random Tree Star algorithm.

Second, gains are largest in structured coordination and communication tasks, where relational structure is explicit. For example, Mou et al. [47] report a connected time-step ratio of 0.311 for the GCN-based CR-MGCM method, compared with 0.200 for the strongest

non-proposed baseline in the same setting. However, the comparison also shows that the apparent improvement depends strongly on which baseline is selected; using the HERO baseline instead gives 0.311 versus 0.077.

Across the fourteen representative studies examined, only four (28.6%) report or permit computation of a relative improvement against a defined baseline, and four (28.6%) report absolute metrics without a baseline; these eight studies with numerical results appear in Table 8. The remaining six (42.9%) report no quantitative performance value and are summarized qualitatively in Table 10. This distribution is itself a finding: the GNN-for-UAV literature currently lacks shared baselines and standardized reporting, which limits reproducibility and makes meta-analysis across studies difficult. Studies that do not report quantitative performance values are summarized qualitatively in Table 10.

Because the studies summarized in Table 8 report heterogeneous metrics across different datasets and application domains, direct numerical ranking may be misleading. Therefore, Table 10 presents a qualitative summary by application domain, model family, evaluation focus, and evidence type rather than comparing raw performance values.

Table 10. Qualitative summary of representative GNN-based UAV and multi-agent studies by application domain, model family, evaluation focus, and evidence type.

Application Domain	Model Family	Evaluation Focus	Evidence Type
Trajectory Planning	GCN + RL/Graph RL	Navigation efficiency, collision avoidance, cooperative trajectory planning	Simulation
Target Tracking	GAT/Cross-Attention	HOTA, MOTA, IDF1, tracking consistency	UAV benchmarks
Communication/FANET	GCN/Graph RL/Meta-learning/GNN + SAC	Connectivity, robustness, throughput, secure communication [147]	Simulation
Task Allocation	GNN + PPO/CTDE/MAGNNET/ GCN/Heterogeneous GNN + RL/Graph RL	Conflict-free assignment, task completion, computation offloading [108,136,148]	Simulation
Visual Tracking and Perception	GNN-enhanced visual models	FPS, tracking accuracy, small-object robustness	Benchmark and real-world data
Fault and Anomaly Detection	ST-GAT/Transformer/Causal GNN	Accuracy, AUC, residual-based detection	Real UAV flight data
Swarm Coordination	Aggregation GNN/GNN + MAPPO/Decentralized GNN policies	Coverage control, distributed coordination, decentralized control [27,35]	Simulation

Note: This table provides qualitative categorization rather than direct numerical ranking because the reviewed studies use heterogeneous datasets, metrics, baselines, and evaluation settings. Studies that do not report quantitative performance values are summarized qualitatively in this table.

Tables 6 and 10 are intended to be read together. Table 6 provides the recommended architecture for each application domain, while Table 10 records the type of evidence supporting each recommendation. Read jointly, they show that the recommendations differ substantially in evidential strength: guidance for trajectory planning and task allocation rests largely on simulation evidence, guidance for target tracking and visual perception is supported by benchmark datasets and limited real-world data, and only the real-world deployment row of Table 10 is supported by physical-robot evidence. Recommendations supported primarily by simulation should therefore be treated as provisional pending hardware validation, as discussed in Section 4.6.

Comparative Analysis of GNN-Based UAV Systems: A cross-analysis of the studies summarized in Tables 6 and 10 reveals several consistent trends regarding the effectiveness and limitations of GNN-based approaches in UAV and multi-agent systems.

First, GNN-based methods demonstrate clear advantages in decentralized coordination tasks, particularly when integrated with reinforcement learning frameworks. Hybrid approaches combining GNNs with multi-agent reinforcement learning paradigms, such as PPO and centralized training with decentralized execution, are frequently applied to trajectory planning, task allocation, communication scheduling, and swarm coordination.

These approaches are effective because graph message passing enables agents to reason over local interaction structures while supporting scalable decentralized decision-making.

Second, attention-based GNN architectures show strong potential in tasks requiring adaptive interaction modeling, including target tracking, communication optimization, and perception-driven decision-making. By dynamically weighting neighboring nodes, these models improve the ability to prioritize reliable or task-relevant information in partially observable and communication-constrained environments. However, their increased computational and communication overhead remains an important deployment concern, especially for dense UAV swarms.

Third, spatio-temporal and hybrid graph models demonstrate strong capability in monitoring and prediction tasks, such as trajectory forecasting, fault detection, anomaly detection, and UAV health monitoring. These models are particularly useful when both relational dependencies and temporal evolution must be modeled jointly. However, their effectiveness depends on the availability of reliable temporal data and sufficient onboard computation.

Despite these advantages, several limitations remain. A large portion of existing work relies on simulation-based validation, with relatively few studies demonstrating large-scale real-world deployment. Furthermore, the lack of standardized UAV datasets and evaluation protocols limits direct comparability across studies. Challenges related to scalability, communication overhead, robustness under latency and packet loss, and embedded hardware constraints remain open research problems.

Overall, GNN-based methods provide a powerful and flexible framework for enabling intelligent, decentralized UAV systems. Future research should focus on bridging the gap between simulation and real-world deployment, developing standardized benchmarks, and improving robustness under practical operational constraints.

3.8. Meta-Summary and Cross-Domain Insights

While Table 8 summarizes individual studies, Table 11 provides a higher-level synthesis of GNN-based UAV applications across major domains. This meta-summary highlights common model families, dataset types, performance trends, and limitations, thereby addressing the need for a more structured cross-study comparison.

Table 11. Meta-summary of GNN-based UAV applications and performance trends across domains.

Application Domain	Typical GNN Model	Primary Task	Dataset Type	Performance Trend	Key Limitation
Trajectory Planning	GNN + Reinforcement Learning (GCN + PPO, Graph RL)	Collision-free navigation and path optimization	Predominantly simulation-based	High success rates typically above 85–95%	Limited real-world safety validation
Target Tracking	Graph Attention Networks (GAT), Heterogeneous GNN	Multi-object tracking and sensor fusion	UAV benchmarks (VisDrone, UAVDT) and simulation	Improved tracking accuracy and identity consistency	Sensitive to occlusion and noisy observations
Task Allocation	GNN + Multi-Agent Reinforcement Learning (MARL), CTDE	Multi-UAV task assignment and scheduling	Simulation environments	High coordination efficiency often exceeding 90%	Scalability and communication overhead
Communication Optimization	Graph RL, Federated GNN, GCN-based models	Scheduling, routing, and resource allocation	Simulation and network scenarios	Improved throughput, connectivity, and robustness	Lack of standardized communication benchmarks
Swarm Coordination	Decentralized GNN, GraphSAGE, GNN + MAPPO	Distributed control, flocking, and patrolling	Simulation and limited real-world robots	Strong robustness under disturbances and failures	Few large-scale real-world deployments
Anomaly and Fault Detection	Spatio-temporal GNN (ST-GNN), GAT + Transformer	Fault detection and anomaly identification	Real UAV flight data	High accuracy often exceeding 96%	Limited availability of public UAV datasets
Security and Defense	GNN + Reinforcement Learning, Graph-based learning	Attack detection, secure communication, recovery	Simulation and graph-based datasets	Effective detection and recovery performance	Limited adversarial and real-world validation

The meta-summary in Table 11 provides a higher-level synthesis of model selection trends across application domains, highlighting how different GNN architectures are preferred depending on task requirements, whereas spatio-temporal and attention-based GNNs are more common in tracking, anomaly detection, and fault diagnosis. Across

domains, GNN-based methods provide clear benefits when the problem involves relational structure, dynamic neighborhood interactions, or decentralized decision-making. Nevertheless, several limitations remain consistent across the literature, including limited real-world validation, heterogeneous evaluation protocols, and insufficient reporting of onboard latency, communication cost, and energy consumption. These observations reinforce the need for standardized UAV-GNN benchmarks and deployment-oriented evaluation criteria.

3.9. Observed Limitations and Inconsistent Performance Trends

While the surveyed studies demonstrate strong performance of GNN-based approaches across various UAV applications, a closer examination of the literature reveals several limitations and inconsistencies.

First, as reflected in Table 8, many studies report improvements over baseline methods without providing standardized or comparable evaluation metrics. In several cases, performance is reported only in relative terms (e.g., percentage improvement) or using task-specific metrics, making it difficult to assess whether GNN-based methods consistently outperform classical approaches across different scenarios.

Second, a number of studies rely on simulation-based environments with controlled conditions. As discussed in Section 4, these settings often simplify communication dynamics, sensing noise, and environmental uncertainty. As a result, the reported performance gains may not directly translate to real-world deployments, where traditional rule-based or optimization-driven methods can exhibit more stable behavior.

Third, the effectiveness of GNN-based models depends heavily on the underlying graph construction and data quality. In scenarios with sparse communication, noisy observations, or rapidly changing topology, the advantages of relational learning may be reduced. This suggests that the performance benefits of GNNs are not uniform across all operational conditions.

Finally, the survey indicates that many works focus on demonstrating feasibility rather than providing rigorous comparisons against strong classical baselines. This limits the ability to draw definitive conclusions about the relative advantages of GNN-based approaches in all UAV applications.

These observations highlight the need for more standardized benchmarking, real-world validation, and systematic comparison with classical methods to fully establish the practical benefits of GNN-based UAV intelligence. In certain scenarios, particularly under strict safety, latency, or resource constraints, simpler optimization-based or rule-based methods may provide more reliable and stable performance than GNN-based approaches.

4. Limitations of Current GNN-Based UAV Approaches

This section critically assesses the limitations of current GNN-based UAV approaches, focusing on methodological constraints, deployment challenges, evaluation gaps, and the transition from simulation to real-world implementation. Although existing studies show promising results across trajectory planning, communication optimization, task allocation, target tracking, anomaly detection, and swarm coordination, practical adoption remains constrained by scalability, communication reliability, computational cost, benchmark inconsistency, and limited safety assurance.

The preceding section established where GNNs help and by how much; this section turns to where they fall short, organized by architecture family so that each limitation can be traced to a specific mechanism.

4.1. Limitations of Convolutional GNNs

Convolutional GNNs, including GCN and GraphSAGE, rely on neighborhood aggregation to propagate information across graph-structured systems [14,33]. While effective for relational representation learning, this mechanism can become restrictive in UAV networks where node interactions are dynamic, heterogeneous, and mission-dependent. In multi-UAV systems, communication quality, spatial proximity, node mobility, and task relevance vary over time, making fixed or weakly adaptive aggregation insufficient to capture critical dependencies.

A key limitation is over-smoothing, where repeated aggregation across multiple layers causes node representations to become increasingly similar. This creates problems for UAV swarms because multi-hop coordination requires information propagation across the network while decentralized decision-making still requires each UAV to preserve distinct local context. Moreover, convolutional GNNs are often developed under relatively stable graph assumptions, whereas UAV networks frequently experience topology changes due to mobility, link disruption, obstacle blockage, and mission-dependent formation changes. As a result, purely convolutional models may struggle to maintain robust performance in real-time UAV operations.

4.2. Limitations of Attention-Based GNNs

Attention-based GNNs, such as Graph Attention Networks, improve upon fixed aggregation by assigning adaptive weights to neighboring nodes [57,135]. This is useful in UAV systems because different neighbors may contribute unequally to routing, collision avoidance, sensing, relay selection, or cooperative tracking. However, attention mechanisms introduce additional computational and communication overhead because attention scores must be computed over graph edges.

This limitation becomes more significant in dense UAV swarms, where the number of communication links can grow rapidly. For time-sensitive UAV tasks such as collision avoidance, formation control, and cooperative navigation, additional processing and message-passing overhead can reduce operational reliability. Attention-based GNNs also assume that messages from neighboring UAVs are available in a timely and reliable manner. In practical FANETs, links may be affected by fading, interference, message loss, and asynchronous communication. Under such conditions, learned attention weights may no longer represent the actual system state, reducing coordination quality. Wireless-aware GNN research highlights the need to explicitly account for channel impairments rather than assuming ideal message exchange [34].

4.3. Limitations of Spatio-Temporal and Dynamic GNNs

Spatio-temporal GNNs extend graph learning by modeling both relational structure and temporal evolution. In UAV and aviation contexts, these models support trajectory prediction, airspace complexity modeling, and dynamic monitoring applications [50,61]. However, they depend strongly on consistent temporal observations and reliable historical data.

In UAV systems, temporal data may be incomplete or noisy due to sensor faults, occlusions, communication interruptions, localization errors, or environmental disturbances. Missing, delayed, or noisy observations can reduce prediction reliability, especially in safety-critical applications such as collision avoidance and cooperative maneuvering. Dynamic graph models also require frequent graph updates to represent changing UAV positions, communication links, and task relationships. These updates introduce additional computational overhead and may be difficult to perform onboard resource-constrained UAVs in real time.

4.4. Limitations of Distributed and Resource-Efficient GNNs

Distributed and resource-efficient GNNs are increasingly important for UAV systems because onboard platforms have limited computation, memory, energy, and communication bandwidth. Distributed GNN training systems and sampling-based methods can improve scalability, but many such methods are developed for general large-scale graph workloads rather than UAV-specific deployment [36,37,149].

Resource-efficient techniques such as quantization, pruning, knowledge distillation, and hardware acceleration can reduce computation and memory requirements [81–84]. However, these methods may introduce information loss or reduce model expressiveness when compression is aggressive. For UAV applications, this creates a trade-off between inference efficiency and mission-level reliability. Lightweight models may be suitable for onboard deployment, but they must still preserve sufficient relational reasoning capacity for navigation, tracking, communication recovery, and safety-critical decision-making.

4.5. Benchmark and Evaluation Limitations

A major limitation in the current literature is the lack of standardized benchmarks for GNN-based UAV systems. Existing studies use different simulators, datasets, assumptions, graph construction strategies, communication models, and evaluation metrics. Some studies assume ideal communication and full observability, while others consider dynamic topology, partial observability, or communication constraints. This makes direct comparison difficult.

Evaluation metrics also vary across studies, including success rate, collision rate, energy consumption, latency, throughput, robustness, tracking accuracy, and task completion efficiency. Without unified benchmark datasets and evaluation protocols, reported improvements may not generalize across UAV platforms, swarm sizes, communication conditions, or mission scenarios. This limitation reduces reproducibility and makes it difficult to identify which GNN architectures are most suitable for specific UAV applications.

4.6. Simulation-to-Real Gap and Deployment Challenges

Many GNN-based UAV methods are evaluated primarily in simulation. Simulated environments often simplify communication, sensing, synchronization, localization, and actuation. Real-world UAV deployments, however, involve asynchronous communication, unreliable wireless links, hardware limitations, sensor noise, localization drift, battery constraints, and environmental uncertainty.

The simulation-to-real gap is especially important for decentralized GNN policies. Real-world multi-robot deployment studies show that decentralized execution requires handling message delays, dropped communication, asynchronous updates, and ad hoc networking constraints [150]. These factors can disrupt message passing and lead to behavior that differs from simulation-trained policies. Therefore, UAV-GNN research needs more real-world testbeds, hardware-in-the-loop experiments, and deployment-aware evaluation.

Because so few studies evaluate the same model in both simulation and real-world settings, simulation results should be read as indicative rather than predictive: they do not always reliably reflect the effectiveness of a method under real-world deployment. The gap is not uniform across tasks or architectures. It is likely widest for methods that depend on dense and timely message passing, such as attention-based coordination and spatio-temporal prediction, because real FANET links introduce latency, packet loss, jitter, and intermittent connectivity. It is comparatively narrower for methods that already assume local, sparse, or asynchronous communication, such as sampling-based and distributed GNN designs. A practical consequence is that an architecture that appears less accurate in idealized simulation, for example one using sparse aggregation, may prove more ro-

bust under deployment conditions. Closing this gap requires shared hardware testbeds, hardware-in-the-loop validation, and a reporting convention in which communication models, latency assumptions, packet-loss rates, localization errors, onboard compute limits, and energy constraints are stated explicitly.

4.7. Safety and Reliability Limitations

Safety remains one of the most critical limitations of GNN-based UAV systems. Many current methods use data-driven learning or reinforcement learning without providing formal guarantees for collision avoidance, stability, constraint satisfaction, or worst-case behavior. This is problematic for safety-critical UAV applications such as airspace management, infrastructure inspection, disaster response, and swarm navigation.

Learned policies may also behave unpredictably outside their training distribution. Dynamic environments, adversarial conditions, sensor failures, or unexpected topology changes can expose weaknesses that were not observed during training. Recent work on causal GNNs, UAV fault detection, anomalous drone detection, and robust swarm communication suggests promising directions for improving reliability, interpretability, and resilience [151–155]. However, formal verification, safety-aware learning, and certified deployment frameworks remain underdeveloped.

4.8. Summary of Limitations

In summary, GNN-based UAV systems face several unresolved limitations. Convolutional GNNs may suffer from over-smoothing and limited adaptability to dynamic topology. Attention-based GNNs improve adaptive aggregation but introduce computational and communication overhead. Spatio-temporal and dynamic GNNs are powerful for modeling temporal evolution but remain sensitive to missing, delayed, or noisy data. Distributed and resource-efficient methods improve scalability but introduce trade-offs between efficiency and model fidelity. In addition, the lack of standardized benchmarks, the simulation-to-real gap, and the absence of formal safety guarantees limit real-world adoption. Addressing these challenges is essential for developing reliable, scalable, and safety-aware GNN-enabled UAV intelligence.

5. Challenges in GNN-Based UAV Systems

Whereas Section 4 examined the limitations specific to each GNN architecture family, this section turns to the system-level challenges that arise when GNN-based methods move from controlled simulation to real UAV deployment. To avoid overlap, neither the architecture-specific limitations of Section 4 nor the architecture comparisons of Section 3.5 and Section 3.6 are repeated here; both are referenced only where a deployment challenge follows directly from them.

Although graph neural networks provide a powerful framework for modeling relational interactions in UAV systems, addressing the limitations identified in the previous section introduces several non-trivial technical challenges. In particular, scalability in large UAV swarms, dynamic topology adaptation, and strict onboard resource constraints pose fundamental obstacles to real-time deployment.

5.1. Scalability Challenges in Large-Scale UAV Networks

When deploying GNNs across large UAV swarms, scalability is a critical concern. As the network grows, computational, communication, and training costs increase substantially.

The main bottleneck is that scalability in UAV systems is not only a graph-size problem; it is also a mobility, communication, and embedded-computation problem. General distributed GNN frameworks typically optimize throughput for large but relatively stable graphs, whereas UAV networks require inference and coordination under continuously

changing topology, intermittent FANET connectivity, limited onboard memory, constrained battery energy, and real-time control deadlines. Sampling-based scalability methods reduce neighborhood size, but if sampling ignores wireless link quality, mission relevance, or safety-critical neighbors, it may remove exactly the information needed for collision avoidance, connectivity preservation, or cooperative tracking. These differences indicate that UAV-oriented scalable GNN design must jointly consider graph sparsification, communication reliability, onboard computation, and mission-level safety rather than adopting general large-scale GNN systems directly.

Distributed GNN Training: Training GNNs on large-scale networks typically requires partitioning graphs into smaller subgraphs that can be processed independently while minimizing communication overhead between partitions [36,67,156]. In UAV networks, spatial locality naturally supports such partitioning since UAVs located near each other form dense local subgraphs. However, high mobility introduces frequent topology changes that require dynamic repartitioning to maintain balanced workloads while minimizing data movement [157]. Distributed frameworks such as DistDGL enable mini-batch training on billion-scale graphs through static partitioning and optimized inter-machine communication [36]. Unlike static data-center environments, UAV networks require mobility-aware partitioning and adaptive load-balancing mechanisms to maintain communication efficiency [1]. Over-the-air aggregation techniques further reduce communication overhead by performing analog gradient aggregation directly within wireless channels [37].

Scalable Training Techniques: Full-batch GNN training requires aggregating information from all k -hop neighbors, which leads to exponential neighborhood growth. Neighbor sampling techniques mitigate this problem by randomly selecting fixed-size neighbor subsets at each layer, reducing computational complexity from exponential to constant per node [143]. Importance sampling further prioritizes neighbors with stronger communication links or greater task relevance [149]. Layer-wise sampling and precomputation strategies reduce computational overhead by caching intermediate representations [141,142]. These large-scale GNN training methods demonstrate transferable scalability principles for handling large graph data, but they are not direct evidence of UAV network deployment [144]. Therefore, their relevance to UAV systems should be interpreted as technical background for scalable graph learning rather than as validated UAV-specific performance.

Resource–Communication Trade-offs: Distributed training introduces fundamental trade-offs between local computation and communication [158]. In bandwidth-limited UAV networks, similar resource–communication trade-offs are expected, but large-scale GNN training studies should be treated as transferable background rather than direct UAV deployment evidence. Gradient compression techniques, including quantization and sparsification, significantly reduce bandwidth requirements [157]. Adaptive strategies dynamically balance these trade-offs according to network conditions [67,156].

DistDGL [36] partitions a static graph for distributed training, and EvolveGCN [67] models temporal evolution over synchronized, complete graph snapshots; in a UAV swarm, the partition is invalidated by mobility, and the snapshots cannot be assembled under intermittent, partially observed FANET links.

Current efforts to bridge this gap remain partial and are largely validated in simulation rather than benchmarked against the general frameworks they adapt, but they indicate several promising directions. UAV-specific studies increasingly favor localized message passing, sparse neighbor selection, and decentralized execution to avoid dependence on global graph knowledge. Federated and distributed learning approaches reduce raw-data sharing and improve privacy, but they still require communication-efficient update mechanisms to operate under FANET constraints. Over-the-air aggregation offers one possible direction by exploiting wireless superposition to reduce gradient-aggregation

overhead, while lightweight and resource-efficient GNN designs address onboard SWaP-C limitations through quantization, pruning, distillation, and shallow architectures.

Dynamic and temporal GNN methods also help address topology variation, but their practical use requires event-triggered updates and latency-aware message passing. Overall, the literature is moving from simply borrowing general scalable GNN techniques toward UAV-aware designs that combine sparse local aggregation, communication-aware learning, dynamic graph adaptation, and hardware-conscious model compression.

Even so, no existing framework jointly satisfies the graph-stability, communication, computation, synchronization, and evaluation requirements summarized in Table 12. The systematic adaptation of general distributed GNN systems to UAV constraints has not yet been investigated as a coherent research problem, which we identify as an open direction for the field.

Table 12. Key discrepancies between general-purpose scalable GNN frameworks and UAV-specific deployment requirements.

Aspect	General Large-Scale GNN Frameworks	UAV-Specific Requirement	Adaptation Bottleneck
Graph structure	Often assume relatively stable or slowly updated graph partitions	UAV graphs change rapidly due to mobility, node failures, and link dynamics	Static partitions become invalid and require costly repartitioning
Communication	Assume reliable server or cluster interconnects	FANET links are intermittent, bandwidth-limited, and affected by interference	Message passing and gradient exchange may be delayed, dropped, or asymmetric
Computation	Designed for CPUs/GPUs with substantial memory and power	UAVs operate under strict SWaP-C and battery constraints	Model depth, neighborhood size, and memory footprint must be limited
Caching/precomputation	Cache node embeddings or graph partitions to improve throughput	Cached representations may become stale under rapid topology change	Cache invalidation and update frequency become mission-critical
Synchronization	Often rely on coordinated mini-batch or synchronized updates	UAVs may operate asynchronously with partial observability	Stale gradients and outdated node states reduce coordination reliability
Evaluation objective	Optimize training throughput, memory use, or graph-processing efficiency	UAV deployment requires safety, latency, connectivity, and energy awareness	System-level metrics must be included beyond accuracy or throughput

5.2. Dynamic Graph Changes and Temporal Adaptation

UAV networks exhibit continuously evolving topologies due to mobility, node arrivals and departures, link failures, and environmental changes. Traditional GNN models designed for static graphs struggle to adapt to these dynamics.

Dynamic GNN Architectures: Dynamic graph neural networks model structural evolution through temporal representations that maintain evolving node embeddings using recurrent units or attention mechanisms [68]. Incremental update techniques modify only the embeddings affected by topology changes rather than recomputing the entire graph representation [65,66]. In UAV networks, when new UAVs join or change positions, only nearby nodes require embedding updates. Continuous-time dynamic graph models also enable event-driven updates triggered by network changes [67,68,156].

Temporal Graph Learning: Learning from temporal graph sequences requires balancing short-term responsiveness with long-term consistency [61,69,70,159]. Combining graph-based spatial modeling with temporal convolution, recurrent architectures, or temporal attention enables models to capture both spatial interactions and temporal evolution. Temporal attention mechanisms further enhance performance by dynamically weighting past observations [69,70].

Real-Time Graph Updates: Operational UAV systems must process graph updates in real time to support situational awareness and rapid decision-making [64,132,148]. However, high update frequencies (typically 10–100 Hz) required for trajectory-planning conflict with the computational limitations of embedded processors. Potential solutions include asynchronous update strategies in which different regions update at different rates depending on local conditions, as well as predictive caching techniques that precompute future states during idle computation cycles [148]. Distributed systems must also maintain consistency despite communication delays and packet loss [64]. Detection mechanisms determine when incremental updates remain sufficient and when full recomputation becomes necessary [132].

Figure 6 illustrates the dynamic evolution of UAV communication graphs, highlighting node arrivals, departures, and edge reconfigurations over time. This demonstrates why static graph assumptions are insufficient for real-world UAV systems. In practice, graph updates may occur at high frequencies due to mobility, link instability, and dynamic mission conditions, creating a tension between model responsiveness and onboard computational feasibility. This challenge motivates the need for dynamic GNN architectures, incremental embedding updates, and asynchronous processing strategies that can preserve situational awareness without requiring full graph recomputation at every update step.

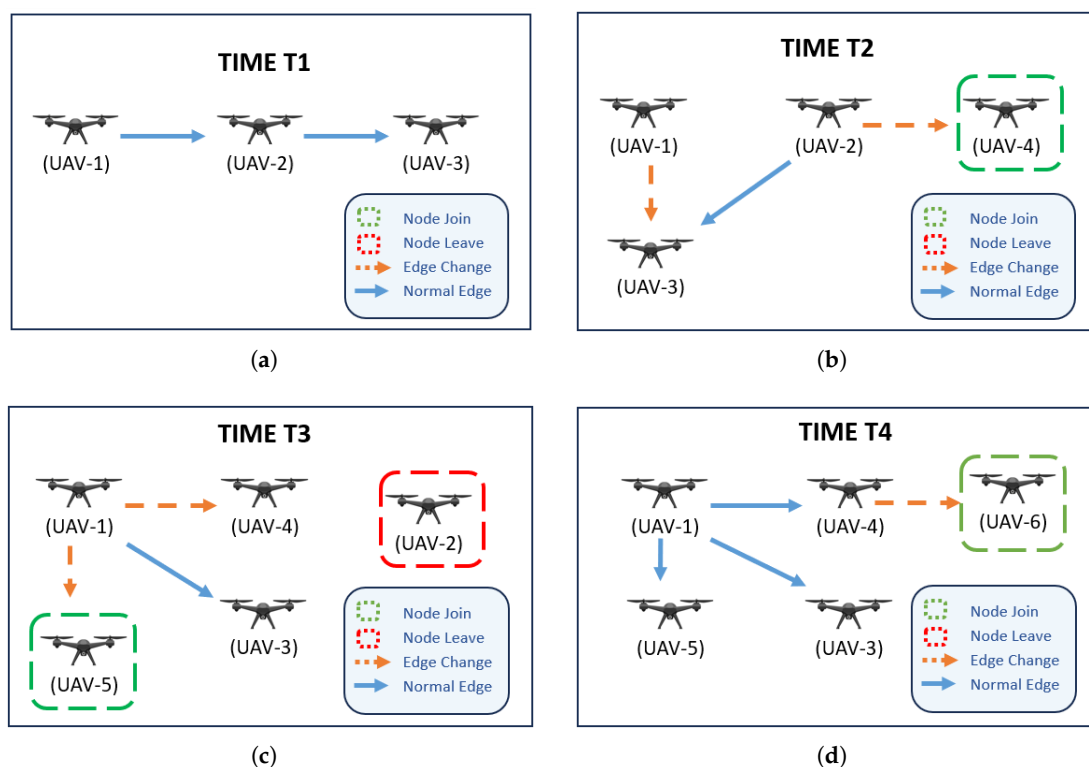


Figure 6. Dynamic evolution of UAV communication graphs over time. (a) Time T1: Initial UAV communication graph. (b) Time T2: Node join (UAV4) and edge changes. (c) Time T3: Node leave (UAV2), node join (UAV5), and edge updates. (d) Time T4: Network stabilization with new node (UAV6).

5.3. Limited Computational Resources Onboard

Small UAV platforms are constrained by strict limits on size, weight, power, and cost (SWaP-C), making the deployment of computationally intensive graph neural network models particularly challenging.

Model Quantization and Compression: Model compression techniques significantly reduce the computational and memory requirements of GNN models, enabling execution

on embedded processors [80–82,111]. Quantization replaces high-precision parameters with lower-precision representations (e.g., 8-bit or 4-bit integers), substantially reducing memory consumption and accelerating inference. For UAV platforms, 8-bit quantization typically reduces model size by approximately 75% with minimal impact on prediction accuracy [80]. Dynamic quantization further adapts numerical ranges based on activation distributions during runtime [82]. Structured pruning techniques remove entire neurons or layers while maintaining a regular network topology, enabling efficient hardware execution.

Hardware Acceleration: Specialized hardware accelerators provide substantial improvements in performance and energy efficiency compared with general-purpose processors [84–86]. These accelerators exploit the irregular sparsity patterns inherent in graph computations through customized memory hierarchies and processing units optimized for gather-scatter operations. FPGA-based accelerators offer flexibility across different GNN architectures while delivering performance improvements of one to two orders of magnitude compared with traditional CPUs [85,86]. Dataflow-based architectures further enhance efficiency by streaming graph data and enabling pipelined execution of communication and computation stages [84].

Lightweight Model Architectures: Designing inherently efficient GNN architectures reduces computational demands from the outset [19]. In many UAV coordination tasks, shallow GNN architectures with two to three layers are sufficient to capture local interaction patterns, reducing computational complexity by an order of magnitude compared with deeper networks. Simple aggregation functions such as mean or max pooling provide low computational overhead but may lose expressive power. More sophisticated aggregators such as attention mechanisms or recurrent units capture richer interaction patterns but increase computational cost [19]. Knowledge distillation techniques transfer information from large teacher models to smaller student models, preserving predictive performance while reducing model size [83].

Federated Learning and Distributed Inference: When individual UAVs lack sufficient computational resources for complex inference tasks, collaborative learning strategies distribute computation across multiple agents [20]. Federated learning allows UAVs to train models collaboratively using locally collected data while sharing only model updates, preserving data privacy and leveraging distributed computational resources [20]. Split-computing approaches further partition GNN inference between UAV edge devices and ground-based cloud infrastructure, balancing latency, bandwidth usage, and computational workload [114].

Energy Efficiency Considerations: Energy consumption directly influences UAV flight duration and mission range [59,112,113]. Adaptive inference techniques can dynamically adjust model complexity based on available battery levels. When sufficient energy is available, full-capacity models may be used; under low-energy conditions, simplified models can be deployed [113]. In UAV-enabled edge computing, model placement, transmission delay, and disk I/O energy cost also affect the feasibility of onboard inference systems [114]. Although this work is not GNN-specific, it provides relevant background on UAV energy constraints that motivate lightweight and resource-efficient learning models. Event-driven processing can further reduce average energy consumption by triggering computations only when significant environmental changes occur [59]. Hardware–software co-design approaches jointly optimize model architecture and hardware implementation to maximize energy efficiency in embedded graph-learning systems [84].

Energy–Latency Trade-offs Across GNN Architectures: Different GNN architectures exhibit distinct trade-offs between computational cost, latency, and energy consumption during inference. Convolution-based models such as GCN and GraphSAGE generally offer lower computational complexity due to simple aggregation operations, making them more

suitable for real-time and energy-constrained UAV applications. In contrast, attention-based models such as GAT introduce additional computational overhead through attention coefficient calculations, increasing inference latency and energy usage, particularly in dense communication graphs.

Spatio-temporal and dynamic GNN models further increase computational requirements by incorporating temporal dependencies and recurrent or sequential processing, which can limit their feasibility for time-critical onboard deployment. While these models provide improved predictive capability and situational awareness, their energy and latency costs must be carefully balanced against system constraints.

These observations highlight that architecture selection in UAV systems involves an inherent trade-off between model expressiveness and resource efficiency, and energy-aware model design remains a key requirement for practical deployment.

Among the challenges summarized in Table 13, dynamic graph adaptation and onboard computational limitations emerge as the most critical barriers to real-world deployment. Although scalability can be improved through graph partitioning, sampling strategies, and distributed training frameworks, real-time UAV operation depends more fundamentally on the ability of models to adapt efficiently to rapidly changing network topologies while remaining computationally feasible on embedded platforms. Consequently, the development of lightweight dynamic graph-learning methods and resource-aware model design should be prioritized.

Table 13. Core Technical Challenges in GNN-Based UAV Systems, Their System-Level Impact, and Representative Solution Strategies.

Challenge Domain	Impact on UAV Systems	Representative Solutions	References
Scalability	Computational and memory complexity increases with graph size, limiting real-time processing in large UAV swarms	Graph partitioning, neighbor sampling, and distributed GNN training frameworks to reduce computational overhead	[36,141–144]
Dynamic Graphs	Rapid topology changes due to UAV mobility lead to unstable message passing and degraded model performance	Temporal and dynamic GNN models that capture evolving graph structures and time-dependent interactions	[65–68,70]
Compute Constraints	Limited onboard processing power, memory, and energy restrict deployment of complex GNN models on UAV platforms	Model quantization, lightweight architectures, hardware acceleration, and distributed/federated inference	[19,20,80,84,86,111]

In addition to these core challenges, several cross-cutting issues further complicate the deployment of GNN-based UAV systems. The limited availability of high-quality, real-world training data motivates the adoption of self-supervised learning and simulation-to-reality transfer techniques.

Simulation-to-real transfer is a further deployment concern and is analyzed in Section 4.6.

Furthermore, achieving reliable generalization across diverse operational environments remains a significant challenge due to variations in communication conditions, sensing noise, and environmental dynamics.

Safety and the role of hybrid learning optimization designs are discussed in Section 4.7 and revisited in the Conclusions Section.

Overall, the challenges discussed in this section show that deployment feasibility depends not only on the selection of an appropriate GNN architecture but also on the ability to manage system-level constraints during operation. Scalability, topology adaptation, and onboard computation are tightly coupled in UAV systems: improving one often affects the

others. Addressing these issues requires coordinated advances in adaptive graph learning, communication-aware model design, and hardware–software co-optimization to support real-time, robust, and energy-efficient UAV intelligence.

6. Future Directions

The future research directions outlined in this section are derived directly from the limitations and challenges identified in Sections 4 and 5, particularly those related to scalability, dynamic graph adaptation, simulation-to-real transfer, energy-constrained deployment, and safety-critical operation. Rather than presenting general trends in UAV or machine learning research, these directions reflect specific gaps observed in the current GNN-based UAV literature. Three promising integration directions are identified based on an analysis of current applications and deployment challenges: reinforcement learning for autonomous decision-making, edge computing for distributed intelligence, and 6G networks for next-generation connectivity. These directions address key system constraints while enabling more practical and scalable UAV deployments. In particular, these directions are motivated by unresolved challenges in dynamic graph adaptation, simulation-to-real transfer, communication reliability, and energy-constrained deployment identified in Section 5.

6.1. Integration with Reinforcement Learning

Motivated by the scalability and coordination challenges discussed in Section 5, the integration of reinforcement learning with GNNs can enable architectures that allow multiple UAVs to make autonomous decisions through interaction with the environment rather than relying solely on predefined rules.

GNN-Enhanced Multi-Agent Reinforcement Learning: In traditional multi-agent reinforcement learning models, increasing the number of agents causes an exponential growth in the state-action space. GNN-based solutions address this challenge by explicitly modeling agent interactions through graph representations [137,138,145,147]. Each agent learns from information within its graph locality, resulting in effective cooperation among agents. Empirical studies show that GNN-based approaches can scale to large agent populations, including swarms of more than 50 UAVs, while maintaining effective learning dynamics [35,139]. The core idea is that GNN-based models naturally fit agent systems with graph structure, where agents act as nodes and their interactions act as edges.

Deep RL Foundations: Basic deep reinforcement learning algorithms such as policy gradient methods, actor–critic architectures, and Q-learning variants form the foundation of GNN-enhanced methods [160–162]. This integration combines the structural inductive bias of GNNs with the capacity of reinforcement learning to optimize long-term objectives through trial-and-error interaction. This is especially useful for UAV applications where optimal behaviors emerge from local interactions.

Graph-Based RL Applications: Several UAV applications show strong potential for GNN-RL integration. In trajectory planning, graph-structured policies support collision avoidance and connectivity maintenance [38,39,41]. In communication network optimization, GNN-RL can jointly optimize routing, power control, and spectrum allocation [26]. In task allocation, graph-based policies support scalable assignment decisions [55,146]. These applications often outperform non-graph baselines while also generalizing better to larger swarm sizes.

Advanced Learning Paradigms: Meta-learning supports rapid adaptation to new mission types with limited retraining. Transfer learning helps bridge the gap between simulation and reality, enabling policies trained in simulation to be deployed on real UAV systems [163–165].

6.2. Integration with Edge Computing

Addressing the computational and communication constraints identified in Section 5, edge computing architectures distribute intelligence across UAV swarms and ground infrastructure. This addresses computational limitations while reducing latency and preserving data privacy.

Federated Learning for UAV Systems: Federated learning enables multiple UAV fleets to collaboratively train GNNs without centralizing sensitive operational data [71,73,76,77,166]. Each UAV trains on locally collected mission data and transmits only model updates rather than raw observations. Privacy-preserving aggregation protocols protect sensitive flight patterns while improving collective intelligence [72]. This paradigm is especially valuable in multi-operator settings such as delivery services, air taxis, and emergency response, where collaboration is required without full data sharing. Combining federated learning with lightweight methods also helps address UAV resource limitations [20].

Over-the-Air Computation: Instead of sending digital gradient updates, over-the-air computation uses wireless channel superposition to aggregate analog signals directly [78,79,167,168]. Multiple UAVs transmit simultaneously, and the wireless medium performs gradient averaging in the electromagnetic domain. This approach significantly reduces communication overhead and enables collaborative learning in near real time during active missions.

Computation Offloading and Edge Intelligence: Dynamic offloading divides GNN execution between UAVs and ground infrastructure in order to balance latency, energy consumption, and inference quality [114,136,169–172]. Lightweight front-end processing on UAVs handles time-sensitive perception, while more complex reasoning is offloaded to edge servers. Topology-aware scheduling strategies account for changes in network conditions [75]. Digital twin frameworks create virtual replicas of UAV systems on edge infrastructure, enabling safe policy validation and testing [104,173,174]. Related 6G-enabled digital-twin routing studies further show that workload prediction and multi-agent reinforcement learning can support adaptive migration and routing decisions in dynamic edge environments [175], offering useful background for future UAV digital-twin and edge-computing architectures.

6.3. Connectivity to 6G Networks

Given the communication latency and reliability challenges highlighted in Section 5, the emerging sixth generation of wireless communication technologies offers several transformational features, including terabit-per-second data rates, sub-millisecond latency, and centimeter-level positioning accuracy. These advances are likely to revolutionize UAV operation by enabling large numbers of autonomous aerial devices to function simultaneously.

6G Architecture and UAV Integration: Next-generation networks naturally incorporate aerial nodes as part of their infrastructure [176–180]. UAVs can act as flying base stations, relay nodes, and sensing platforms. In highly mobile aerial-terrestrial networks, GNN-based network management can optimize three-dimensional cell deployment, handover strategies, and interference coordination. Recent studies have begun exploring GNN-based modeling for heterogeneous multi-tier networks, including UAV, high-altitude platform (HAP), and satellite integration in 6G-enabled non-terrestrial networks [10,16].

Advanced 6G Technologies: Non-terrestrial networks supported by satellites provide service to remote areas [181]. Terahertz communication enables ultra-high-speed links for streaming sensor data [182]. Ultra-reliable low-latency communication supports stringent control requirements by minimizing end-to-end delay [31,32]. AI-native network management can dynamically allocate resources through GNN-based controllers [183]. The Internet of Things provides a foundation for large-scale UAV sensing ecosystems [184].

Recent 6G-enabled digital-twin migration studies also suggest that workload prediction and multi-agent reinforcement learning can support adaptive routing and service migration in highly dynamic edge networks [175].

6.4. Emerging Directions

In addition to these three primary integration areas, and building on the limitations and challenges identified in Sections 4 and 5, several additional research directions emerge that warrant further investigation. These include adversarial robustness and security mechanisms that protect UAV systems against attacks [185–189]; fault detection and resilience frameworks that ensure safe degraded-mode operation [151–155,190–192]; neural architecture search methods that automate the design of suitable GNN architectures for specific UAV tasks [193,194]; and cross-domain applications that extend GNN-UAV techniques to ground robotics, maritime systems, and satellite-like distributed autonomous networks [195–201].

Furthermore, competitive and adversarial multi-UAV scenarios are gaining attention, where multiple agents may operate with conflicting objectives in shared environments. Graph neural networks provide a natural framework to model such interactions, enabling scalable coordination, opponent-aware strategies, and robust decision-making under adversarial conditions.

Taken together, these directions indicate a transition toward UAV systems that are increasingly autonomous, scalable, and communication-aware through the integration of GNNs with complementary technologies. However, translating these advances into practical deployments requires addressing system-level constraints identified in Section 5, particularly those related to communication reliability, computational efficiency, and real-time operation.

Toward Deployment-Ready GNN Frameworks: A key research direction is the development of deployment-aware GNN frameworks that explicitly incorporate communication constraints, energy efficiency, and latency requirements. In practical UAV networks, intermittent connectivity and limited bandwidth necessitate adaptive message-passing mechanisms and communication-efficient learning strategies. Integrating GNNs with edge computing and distributed learning paradigms can further improve scalability and enable real-time deployment in large-scale UAV systems [20,34,71].

Standardized Benchmarking and Real-World Validation: Improving reproducibility and comparability across studies requires standardized benchmarks, datasets, and evaluation protocols tailored to UAV systems. Most existing works rely on simulation environments that do not fully capture real-world variability. Future research should therefore emphasize hardware-in-the-loop experiments and real-world validation to bridge the gap between simulation and deployment [1,18].

Safety-Critical and Robust Learning: Ensuring robustness and reliability remains a fundamental requirement for real-world UAV applications. Models must operate under uncertainty, noisy observations, and dynamic environments while maintaining stable performance. Future work should focus on uncertainty-aware learning, robust training strategies, and resilience to communication failures. In particular, approaches that explicitly model dynamic graph evolution and decentralized coordination can improve reliability in practical operating conditions [67,68,70].

In addition, recent research has explored integrating learning-based policies with formal safety guarantees, such as control barrier functions (CBFs) and constraint-aware optimization. Combining graph neural networks with such frameworks enables decentralized decision-making while ensuring provable safety and constraint satisfaction in multi-UAV systems.

6.5. Visionary Research Opportunities

Beyond the integration directions discussed above, the next stage of GNN-enabled UAV research depends on mechanisms that directly target current deployment barriers. Table 14 identifies six such opportunities, each characterized by its technical direction, the UAV-specific gap it addresses, and the expected research outcome.

Table 14. Visionary and technically detailed research opportunities for GNN-enabled UAV systems.

Research Opportunity	Technical Direction	UAV Gap Addressed	Expected Research Outcome
Event-triggered GNN message passing	Activate graph communication only when topology, uncertainty, or mission risk crosses a threshold	High communication overhead and energy use in dense swarms	Communication-efficient GNNs with bounded latency and lower battery cost
Safety-certified GNN policies	Combine GNN policies with control barrier functions, model predictive control, or reachability analysis	No formal guarantees for collision avoidance or safe control	Coordination with verifiable safety constraints
Continual topology-adaptive GNNs	Online graph rewiring, memory modules, and uncertainty-aware updates for changing networks	Static training fails under node failure, packet loss, and changing membership	GNNs that adapt during deployment without full retraining
Hardware-aware GNN co-design	Jointly optimize depth, neighborhood size, quantization, and onboard processor constraints	Many GNNs are too costly for real-time embedded deployment	Deployable models optimized for latency, memory, and energy
Sim-to-real graph digital twins	Digital twins, domain randomization, and hardware-in-the-loop validation of learned policies	Simulation results often do not transfer to real flight	Reproducible validation pipeline for deployment-ready systems
Trust-aware, attack-resilient graph learning	Edge trust scores, anomaly detection, and adversarially robust aggregation	Vulnerability to spoofed nodes, corrupted links, and malicious data	Robust coordination under cyber–physical attacks

7. Conclusions

This survey shows that the effectiveness of GNN-based UAV intelligence depends less on choosing a single dominant model family and more on aligning architecture design with mission constraints. The main synthesis emerging from the reviewed literature is that different GNN families address different UAV requirements. Convolutional GNNs and GraphSAGE-style models are most suitable for large-scale swarm coordination because they support efficient local message passing and scalability. However, they are less effective when communication quality, sensing reliability, or agent roles vary significantly across the network. In contrast, attention-based GNNs are more suitable for heterogeneous sensing, cooperative target tracking, and adaptive neighbor weighting, but their computational overhead and sensitivity to noisy communication make real-world deployment more challenging.

A second important finding is that the largest deployment gap appears in models that depend on dynamic graph updates, temporal sequences, or reinforcement learning policies. Spatio-temporal and dynamic GNNs are promising for trajectory prediction, FANET adaptation, and airspace monitoring, but their performance depends heavily

on reliable temporal observations and frequent graph updates. Similarly, GNN-MARL frameworks show strong simulation performance for coordination and task allocation, but they remain difficult to validate under packet loss, asynchronous communication, partial observability, and onboard computation limits. These observations indicate that simulation accuracy alone is insufficient for judging deployment readiness.

A third cross-cutting insight is that hybrid approaches are likely to be the most practical near-term direction for safety-critical UAV systems. Pure learning-based methods provide flexibility and fast inference, while classical optimization and control methods provide constraint handling and interpretability. Combining GNNs with optimization, model predictive control, safety filters, or rule-based fallback mechanisms can better balance adaptability, efficiency, and reliability in real UAV missions.

The most promising near-term research directions can therefore be ranked as follows. First, lightweight dynamic GNNs should be prioritized because UAV systems require models that can adapt to changing topology while operating under strict onboard energy and computation limits. Second, communication-aware distributed and federated GNNs are essential for scalable swarm deployment, especially when centralized coordination is infeasible. Third, simulation-to-real transfer, hardware-in-the-loop validation, and real-flight benchmarking are needed to close the gap between controlled experiments and operational UAV environments. Fourth, safety-aware hybrid GNN frameworks should be developed to provide stronger guarantees for collision avoidance, formation control, and mission-critical decision making.

For practitioners, GNN architecture selection should begin with the UAV mission requirement rather than model popularity. For large-scale coordination with limited computation, GCN or GraphSAGE-style models are appropriate starting points. For target tracking, heterogeneous sensing, or variable-quality communication links, attention-based or heterogeneous GNNs are better suited. For trajectory forecasting, evolving FANETs, and airspace prediction, spatio-temporal or dynamic GNNs should be considered, provided that temporal data quality and update latency can be managed. For embedded UAV deployment, distributed and resource-efficient variants should be treated as necessary design constraints rather than optional enhancements.

Overall, GNN-enabled UAV intelligence is moving from task-specific simulation studies toward deployment-aware system design. Future progress will depend on models that are not only accurate, but also communication-efficient, energy-aware, robust to topology changes, and compatible with safety-critical UAV operation.

Author Contributions: Conceptualization, R.P. and A.S.; Methodology, R.P. and A.S.; Investigation, R.P.; Formal analysis, R.P.; Writing—original draft preparation, R.P.; Writing—review and editing, R.P. and A.S.; Visualization, R.P.; Supervision, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Representative Search Queries

The representative query below illustrates the search logic, which was adapted to the native syntax of each database listed in Section 1.5.

Core query:

("Graph Neural Network" OR "GNN" OR "Graph Convolutional Network" OR "GAT" OR "GraphSAGE" OR "Spatio-Temporal Graph") AND ("UAV" OR "Unmanned Aerial

Vehicle" OR "Drone" OR "FANET" OR "UAV swarm") AND ("trajectory" OR "tracking" OR "communication" OR "coordination" OR "task allocation")

- IEEE Xplore, ACM Digital Library: the Boolean query above applied to all-metadata/full-text fields.
- ScienceDirect, SpringerLink, MDPI: the same keyword groups applied to title, abstract, and keywords, with the Boolean operators adapted to each interface.
- arXiv: the keyword groups applied to title and abstract (categories cs.LG, cs.RO, eess.SP).
- Google Scholar: shorter combined-term queries (e.g., "GNN UAV," "graph neural network drone," "graph-based UAV coordination," "GNN FANET routing") to capture records not indexed in the databases above.

References

1. Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.-H.; Debbah, M. A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2334–2360.
2. Shakhathreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634.
3. Mohsan, S.A.; Khan, M.A.; Noor, F.; Ullah, I.; Alsharif, M.H. Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review. *Drones* **2022**, *6*, 147.
4. Bu, Y.; Yan, Y.; Yang, Y. Advancement Challenges in UAV Swarm Formation Control: A Comprehensive Review. *Drones* **2024**, *8*, 320.
5. Alqudsi, Y.; Makaraci, M. UAV Swarms: Research, Challenges, and Future Directions. *J. Eng. Appl. Sci.* **2025**, *72*, 12.
6. Chung, S.J.; Paranjape, A.A.; Dames, P.; Shen, S.; Kumar, V. A Survey on Aerial Swarm Robotics. *IEEE Trans. Robot.* **2018**, *34*, 837–855.
7. Tahir, A.; Böling, J.; Haghbayan, M.H.; Toivonen, H.T.; Plosila, J. Swarms of Unmanned Aerial Vehicles—A Survey. *J. Ind. Inf. Integr.* **2019**, *16*, 100106.
8. Cao, Y.; Yu, W.; Ren, W.; Chen, G. An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination. *IEEE Trans. Ind. Inform.* **2013**, *9*, 427–438.
9. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4–24.
10. He, S.; Xiong, S.; Ou, Y.; Zhang, J.; Wang, J.; Huang, Y.; Zhang, Y. An Overview on the Application of Graph Neural Networks in Wireless Networks. *IEEE Open J. Commun. Soc.* **2021**, *2*, 2547–2565. <https://doi.org/10.1109/OJCOMS.2021.3128637>.
11. Jiang, W. Graph-Based Deep Learning for Communication Networks: A Survey. *Comput. Commun.* **2022**, *185*, 40–54. <https://doi.org/10.1016/j.comcom.2021.12.015>.
12. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The Graph Neural Network Model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80. <https://doi.org/10.1109/TNN.2008.2005605>.
13. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph Neural Networks: A Review of Methods and Applications. *AI Open* **2020**, *1*, 57–81.
14. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017. Available online: <https://openreview.net/forum?id=SJU4ayYgl> (accessed on 6 January 2026).
15. Suárez-Varela, J.; Almasan, P.; Ferriol-Galmés, M.; Rusek, K.; Geyer, F.; Cheng, X.; Shi, X.; Xiao, S.; Scarselli, F.; Cabellos-Aparicio, A.; Barlet-Ros, P. Graph Neural Networks for Communication Networks: Context, Use Cases and Opportunities. *IEEE Netw.* **2023**, *37*, 146–153. <https://doi.org/10.1109/MNET.123.2100773>.
16. Shen, Y.; Zhang, J.; Song, S.H.; Letaief, K.B. Graph Neural Networks for Wireless Communications: From Theory to Practice. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 3554–3569. <https://doi.org/10.1109/TWC.2022.3219840>.
17. Lu, Y.; Li, Y.; Zhang, R.; Chen, W.; Ai, B.; Niyato, D. Graph Neural Networks for Wireless Networks: Graph Representation, Architecture and Evaluation. *IEEE Wirel. Commun.* **2025**, *32*, 150–156. <https://doi.org/10.1109/MWC.006.2400131>.
18. Zeng, Y.; Zhang, R.; Lim, T.J. Wireless Communications with Unmanned Aerial Vehicles: Opportunities and Challenges. *IEEE Commun. Mag.* **2016**, *54*, 36–42. <https://doi.org/10.1109/MCOM.2016.7470933>.
19. Kose, H.T.; Nunez-Yanez, J.; Piechocki, R.; Pope, J. A Survey of Computationally Efficient Graph Neural Networks for Reconfigurable Systems. *Information* **2024**, *15*, 377. <https://doi.org/10.3390/info15070377>.

20. Zhong, L.; Chen, Z.; Cheng, H.; Li, J. Lightweight Federated Graph Learning for Accelerating Classification Inference in UAV-Assisted MEC Systems. *IEEE Internet Things J.* **2024**, *11*, 21180–21190. <https://doi.org/10.1109/JIOT.2024.3365675>.
21. Singh, H.; Verma, S.; Adhya, A. A Brief Review on Position-Based Routing Protocols in Flying Ad-Hoc Networks. In *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*; IEEE: New York, NY, USA, 2022; pp. 1423–1428. <https://doi.org/10.1109/IC3I56241.2022.10072865>.
22. Amponis, G.; Lagkas, T.; Sarigiannidis, P.; Vitsas, V.; Fouliras, P.; Wan, S. A Survey on FANET Routing from a Cross-Layer Design Perspective. *J. Syst. Archit.* **2021**, *120*, 102281. <https://doi.org/10.1016/j.sysarc.2021.102281>.
23. Khan, M.F.; Yau, K.L.A.; Noor, R.M.; Imran, M.A. Routing Schemes in FANETs: A Survey. *Sensors* **2020**, *20*, 38. <https://doi.org/10.3390/s20010038>.
24. Gupta, L.; Jain, R.; Vaszkun, G. Survey of Important Issues in UAV Communication Networks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1123–1152.
25. Pan, Y.; Wang, X.; Xu, Z.; Cheng, N.; Xu, W.; Zhang, J.J. GNN-Empowered Effective Partial Observation MARL Method for AoI Management in Multi-UAV Network. *IEEE Internet Things J.* **2024**, *11*, 34541–34553. <https://doi.org/10.1109/JIOT.2024.3447774>.
26. Ye, Z.; Wang, K.; Chen, Y.; Jiang, X.; Song, G. Multi-UAV Navigation for Partially Observable Communication Coverage by Graph Reinforcement Learning. *IEEE Trans. Mob. Comput.* **2023**, *22*, 4056–4069. <https://doi.org/10.1109/TMC.2022.3146881>.
27. Tolstaya, E.; Gama, F.; Paulos, J.; Pappas, G.; Kumar, V.; Ribeiro, A. Learning Decentralized Controllers for Robot Swarms with Graph Neural Networks. In *Proceedings of the Conference on Robot Learning*; PMLR: Cambridge, MA, USA, 2020; pp. 671–682.
28. Aguzzi, G.; Domini, D.; Venturini, F.; Viroli, M. Scaling Swarm Coordination with GNNs—How Far Can We Go? *AI* **2025**, *6*, 282. <https://doi.org/10.3390/ai6110282>.
29. Pedro, D.; Matos-Carvalho, J.P.; Fonseca, J.M.; Mora, A. Collision Avoidance on Unmanned Aerial Vehicles Using Neural Network Pipelines and Flow Clustering Techniques. *Remote Sens.* **2021**, *13*, 2643. <https://doi.org/10.3390/rs13132643>.
30. Gräfe, A.; Eickhoff, J.; Trimpe, S. Event-Triggered and Distributed Model Predictive Control for Guaranteed Collision Avoidance in UAV Swarms. *IFAC-PapersOnLine* **2022**, *55*, 79–84. <https://doi.org/10.1016/j.ifacol.2022.07.239>.
31. Masaracchia, A.; Li, Y.; Nguyen, K.K.; Yin, C.; Khosravirad, S.R.; Da Costa, D.B.; Duong, T.Q. UAV-Enabled Ultra-Reliable Low-Latency Communications for 6G: A Comprehensive Survey. *IEEE Access* **2021**, *9*, 137338–137352.
32. Azari, M.M.; Rosas, F.; Chen, K.-C.; Pollin, S. Ultra Reliable UAV Communication Using Altitude and Cooperation Diversity. *IEEE Trans. Commun.* **2018**, *66*, 330–344. <https://doi.org/10.1109/TCOMM.2017.2746105>.
33. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive Representation Learning on Large Graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. Available online: <https://doi.org/10.48550/arXiv.1706.02216> (accessed on 6 January 2026).
34. Gao, Z.; Gündüz, D. Graph Neural Networks over the Air for Decentralized Tasks in Wireless Networks. *IEEE Trans. Signal Process.* **2025**, *73*, 721–737. <https://doi.org/10.1109/TSP.2025.3534685>.
35. Goeckner, A.; Sui, Y.; Martinet, N.; Li, X.; Zhu, Q. Graph Neural Network-Based Multi-Agent Reinforcement Learning for Resilient Distributed Coordination of Multi-Robot Systems. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: New York, NY, USA, 2024; pp. 5732–5739. <https://doi.org/10.1109/IROS58592.2024.10802510>.
36. Zheng, D.; Ma, C.; Wang, M.; Zhou, J.; Su, Q.; Song, X.; Gan, Q.; Zhang, Z.; Karypis, G. DistDGL: Distributed Graph Neural Network Training for Billion-Scale Graphs. In *2020 IEEE/ACM 10th Workshop on Irregular Applications: Architectures and Algorithms (IA3)*; IEEE: New York, NY, USA, 2020; pp. 36–44. <https://doi.org/10.1109/IA351965.2020.00011>.
37. Mostafa, H. Sequential Aggregation and Rematerialization: Distributed Full-Batch Training of Graph Neural Networks on Large Graphs. *Proc. Mach. Learn. Syst.* **2022**, *4*, 265–275.
38. Du, Y.; Qi, N.; Li, X.; Xiao, M.; Boulogeorgos, A.-A.A.; Tsiftsis, T.A.; Wu, Q. Distributed Multi-UAV Trajectory Planning for Downlink Transmission: A GNN-Enhanced DRL Approach. *IEEE Wirel. Commun. Lett.* **2024**, *13*, 3578–3582. <https://doi.org/10.1109/LWC.2024.3480152>.
39. Feng, Z.; Wu, D.; Huang, M.; Yuen, C. Graph-Attention-Based Reinforcement Learning for Trajectory Design and Resource Assignment in Multi-UAV-Assisted Communication. *IEEE Internet Things J.* **2024**, *11*, 27421–27434. <https://doi.org/10.1109/JIOT.2024.3397823>.
40. Westheider, J.; Rückin, J.; Popović, M. Multi-UAV Adaptive Path Planning Using Deep Reinforcement Learning. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: New York, NY, USA, 2023; pp. 649–656. <https://doi.org/10.1109/IROS55552.2023.10342516>.
41. Yang, Q.; Lu, J.; Zhang, Y.; Shao, S. Multi-UAV Cooperative Path Planning via Graph Neural Network and Proximal Policy Optimization. *IEEE Access* **2025**, *13*, 193575–193588. <https://doi.org/10.1109/ACCESS.2025.3629967>.
42. Dabbabi, K.; Delleji, T. Graph Neural Network-Tracker: A Graph Neural Network-Based Multi-Sensor Fusion Framework for Robust Unmanned Aerial Vehicle Tracking. *Vis. Comput. Ind. Biomed. Art* **2025**, *8*, 18. <https://doi.org/10.1186/s42492-025-00200-2>.
43. Zhou, W.; Li, J.; Liu, Z.; Shen, L. Improving Multi-Target Cooperative Tracking Guidance for UAV Swarms Using Multi-Agent Reinforcement Learning. *Chin. J. Aeronaut.* **2022**, *35*, 100–112. <https://doi.org/10.1016/j.cja.2021.09.008>.

44. Qian, H.; Sun, X.; Guo, R.; Su, S.; Ding, B.; Guo, X. Low-Altitude Multi-Object Tracking via Graph Neural Networks with Cross-Attention and Reliable Neighbor Guidance. *Remote Sens.* **2025**, *17*, 3502. <https://doi.org/10.3390/rs17203502>.
45. Xia, Z.; Du, J.; Wang, J.; Jiang, C.; Ren, Y.; Li, G.; Han, Z. Multi-Agent Reinforcement Learning Aided Intelligent UAV Swarm for Target Tracking. *IEEE Trans. Veh. Technol.* **2022**, *71*, 931–945. <https://doi.org/10.1109/TVT.2021.3129504>.
46. Wu, D.; Cao, Z.; Lin, X.; Shu, F.; Feng, Z. A Learning-Based Cooperative Navigation Approach for Multi-UAV Systems Under Communication Coverage. *IEEE Trans. Netw. Sci. Eng.* **2025**, *12*, 763–773. <https://doi.org/10.1109/TNSE.2024.3517872>.
47. Mou, Z.; Gao, F.; Liu, J.; Wu, Q. Resilient UAV Swarm Communications with Graph Convolutional Neural Network. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 393–411. <https://doi.org/10.1109/JSAC.2021.3126047>.
48. Xing, N.; Zhang, Y.; Wang, Y.; Zhou, Y. Unmanned Aerial Vehicle Cooperative Data Dissemination Based on Graph Neural Networks. *Sensors* **2024**, *24*, 887. <https://doi.org/10.3390/s24030887>.
49. Shen, X.; Chen, J.; Yan, R. A Spatial-Temporal Model for Network-Wide Flight Delay Prediction Based on Federated Learning. *Appl. Soft Comput.* **2024**, *154*, 111380. <https://doi.org/10.1016/j.asoc.2024.111380>.
50. Li, B.; Li, Z.; Chen, J.; Yan, Y.; Lv, Y.; Du, W. MAST-GNN: A Multimodal Adaptive Spatio-Temporal Graph Neural Network for Airspace Complexity Prediction. *Transp. Res. Part C Emerg. Technol.* **2024**, *160*, 104521. <https://doi.org/10.1016/j.trc.2024.104521>.
51. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 3848–3858. <https://doi.org/10.1109/TITS.2019.2935152>.
52. Zhang, S.; Tong, H.; Xu, J.; Maciejewski, R. Graph Convolutional Networks: A Comprehensive Review. *Comput. Soc. Netw.* **2019**, *6*, 11. <https://doi.org/10.1186/s40649-019-0069-y>.
53. Reynolds, C.W. Flocks, Herds and Schools: A Distributed Behavioral Model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*; Association for Computing Machinery: New York, NY, USA, 1987; pp. 25–34.
54. Gori, M.; Monfardini, G.; Scarselli, F. A New Model for Learning in Graph Domains. In *2005 IEEE International Joint Conference on Neural Networks*; IEEE: New York, NY, USA, 2005; Volume 2, pp. 729–734.
55. Wu, H.; Tian, L.; Tang, H.; Li, R.; Jiao, P. Graph Convolutional Reinforcement Learning-Guided Joint Trajectory Optimization and Task Offloading for Aerial Edge Computing. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 17487–17498. <https://doi.org/10.1109/TITS.2024.3490533>.
56. Chiang, W.-L.; Liu, X.; Si, S.; Li, Y.; Bengio, S.; Hsieh, C.-J. Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 257–266. <https://doi.org/10.1145/3292500.3330925>.
57. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph Attention Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, 30 April–3 May 2018.
58. Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; Yu, P.S. Heterogeneous Graph Attention Network. In *Proceedings of the World Wide Web Conference*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 2022–2032. <https://doi.org/10.1145/3308558.3313562>.
59. Wu, Q.; Liu, Q.; Zhu, W.; Wu, Z. Energy Efficient UAV-Assisted IoT Data Collection: A Graph-Based Deep Reinforcement Learning Approach. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 6082–6094. <https://doi.org/10.1109/TNSM.2024.3450964>.
60. Zhao, B.; Huo, M.; Li, Z.; Yu, Z.; Qi, N. Graph-Based Multi-Agent Reinforcement Learning for Large-Scale UAV Swarm System Control. *Aerosp. Sci. Technol.* **2024**, *150*, 109166. <https://doi.org/10.1016/j.ast.2024.109166>.
61. Seo, Y.; Defferrard, M.; Vandergheynst, P.; Bresson, X. Structured Sequence Modeling with Graph Convolutional Recurrent Networks. In *International Conference on Neural Information Processing*; Springer: Cham, Switzerland, 2018; pp. 362–373. https://doi.org/10.1007/978-3-030-04167-0_33.
62. Chen, S.; Sun, Y.; Li, P.; Zhou, L.; Lu, C.-T. Learning Decentralized Flocking Controllers with Spatio-Temporal Graph Neural Network. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2024; pp. 2596–2602. <https://doi.org/10.1109/ICRA57147.2024.10610627>.
63. Singh, V.; Sahana, S.K.; Bhattacharjee, V. Integrated Spatio-Temporal Graph Neural Network for Traffic Forecasting. *Appl. Sci.* **2024**, *14*, 11477. <https://doi.org/10.3390/app142411477>.
64. Ma, W.; Chu, Z.; Chen, H.; Li, M. Spatio-Temporal Evolutional Graph Neural Network for Traffic Flow Prediction in UAV-Based Urban Traffic Monitoring System. *Sci. Rep.* **2024**, *14*, 26800. <https://doi.org/10.1038/s41598-024-78335-0>.
65. Feng, Z.; Wang, R.; Wang, T.; Song, M.; Wu, S.; He, S. A Comprehensive Survey of Dynamic Graph Neural Networks: Models, Frameworks, Benchmarks, Experiments and Challenges. *IEEE Trans. Knowl. Data Eng.* **2026**, *38*, 26–46. <https://doi.org/10.1109/TKDE.2025.3621291>.
66. Jin, M.; Koh, H.Y.; Wen, Q.; Zambon, D.; Alippi, C.; Webb, G.I.; King, I.; Pan, S. A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 10466–10485. <https://doi.org/10.1109/TPAMI.2024.3443141>.

67. Pareja, A.; Domeniconi, G.; Chen, J.; Ma, T.; Suzumura, T.; Kanezashi, H.; Kaler, T.; Schardl, T.; Leiserson, C. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 5363–5370. <https://doi.org/10.1609/aaai.v34i04.5984>.
68. Manessi, F.; Rozza, A.; Manzo, M. Dynamic Graph Convolutional Networks. *Pattern Recognit.* **2020**, *97*, 107000. <https://doi.org/10.1016/j.patcog.2019.107000>.
69. Kumar, S.; Zhang, X.; Leskovec, J. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1269–1278. <https://doi.org/10.1145/3292500.3330895>.
70. Kazemi, S.M.; Goel, R.; Jain, K.; Kobyzev, I.; Sethi, A.; Forsyth, P.; Poupart, P. Representation Learning for Dynamic Graphs: A Survey. *J. Mach. Learn. Res.* **2020**, *21*, 1–73. <http://jmlr.org/papers/v21/19-447.html>.
71. Liu, R.; Xing, P.; Deng, Z.; Li, A.; Guan, C.; Yu, H. Federated Graph Neural Networks: Overview, Techniques, and Challenges. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 4279–4295. <https://doi.org/10.1109/TNNLS.2024.3360429>.
72. Wang, Z.; Zhou, Y.; Shi, Y. Over-the-Air Federated Graph Learning. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 18669–18683. <https://doi.org/10.1109/TWC.2024.3471906>.
73. Wu, C.; Wu, F.; Lyu, L.; Qi, T.; Huang, Y.; Xie, X. A Federated Graph Neural Network Framework for Privacy-Preserving Personalization. *Nat. Commun.* **2022**, *13*, 3091. <https://doi.org/10.1038/s41467-022-30714-9>.
74. Gao, Z.; Gündüz, D. AirGNNs: Graph Neural Networks over the Air. In *2023 31st European Signal Processing Conference (EUSIPCO)*; IEEE: New York, NY, USA, 2023; pp. 963–967. <https://doi.org/10.23919/EUSIPCO58844.2023.10289977>.
75. Wu, J.; Dong, F.; Leung, H.; Zhu, Z.; Zhou, J.; Drew, S. Topology-Aware Federated Learning in Edge Computing: A Comprehensive Survey. *ACM Comput. Surv.* **2024**, *56*, 262. <https://doi.org/10.1145/3659205>.
76. Cheriguene, Y.; Jaafar, W.; Yanikomeroglu, H. Federated Learning in UAV-Assisted MEC Systems: A Comprehensive Survey. *IEEE Open J. Commun. Soc.* **2025**, *6*, 7645–7676. <https://doi.org/10.1109/OJCOMS.2025.3608657>.
77. Fu, M.; Shi, Y.; Zhou, Y. Federated Learning via Unmanned Aerial Vehicle. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 2884–2900. <https://doi.org/10.1109/TWC.2023.3303492>.
78. Guo, H.; Zhu, Y.; Ma, H.; Lau, V.K.; Huang, K.; Li, X.; Nong, H.; Zhou, M. Over-the-Air Aggregation for Federated Learning: Waveform Superposition and Prototype Validation. *J. Commun. Inf. Netw.* **2021**, *6*, 429–442.
79. Hong, J.P.; Park, S.; Choi, W. Over-the-Air Aggregation-Based Federated Learning in Cache-Enabled Wireless Edge Networks. In *ICC 2023-IEEE International Conference on Communications*; IEEE: New York, NY, USA, 2023; pp. 1994–1999. <https://doi.org/10.1109/ICC45041.2023.10279408>.
80. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2018; pp. 2704–2713. <https://doi.org/10.1109/CVPR.2018.00286>.
81. Bahri, M.; Bahl, G.; Zafeiriou, S. Binary Graph Neural Networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2021; pp. 9492–9501. <https://doi.org/10.1109/CVPR46437.2021.00937>.
82. Chen, S.; Zheng, D.; Ding, C.; Huan, C.; Ji, Y.; Liu, H. TANGO: Re-Thinking Quantization for Graph Neural Network Training on GPUs. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 1–15. <https://doi.org/10.1145/3581784.3607047>.
83. Yang, Y.; Qiu, J.; Song, M.; Tao, D.; Wang, X. Distilling Knowledge from Graph Convolutional Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2020; pp. 7074–7083. <https://doi.org/10.1109/CVPR42600.2020.00710>.
84. Geng, T.; Li, A.; Shi, R.; Wu, C.; Wang, T.; Li, Y.; Haghi, P.; Tumeo, A.; Che, S.; Reinhardt, S.; Herbordt, M.C. AWB-GCN: A Graph Convolutional Network Accelerator with Runtime Workload Rebalancing. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*; IEEE: New York, NY, USA, 2020; pp. 922–936. <https://doi.org/10.1109/MICRO50266.2020.00079>.
85. Yan, M.; Deng, L.; Hu, X.; Liang, L.; Feng, Y.; Ye, X.; Zhang, Z.; Fan, D.; Xie, Y. HyGCN: A GCN Accelerator with Hybrid Architecture. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*; IEEE: New York, NY, USA, 2020; pp. 15–29. <https://doi.org/10.1109/HPCA47549.2020.00012>.
86. Ham, T.J.; Wu, L.; Sundaram, N.; Satish, N.; Martonosi, M. Graphicionado: A High-Performance and Energy-Efficient Accelerator for Graph Analytics. In *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*; IEEE: New York, NY, USA, 2016; pp. 1–13. <https://doi.org/10.1109/MICRO.2016.7783759>.
87. Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; Liu, T.-Y. Do Transformers Really Perform Badly for Graph Representation? *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 28877–28888.
88. Rampášek, L.; Galkin, M.; Dwivedi, V.P.; Luu, A.T.; Wolf, G.; Beaini, D. Recipe for a General, Powerful, Scalable Graph Transformer. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 14501–14511.

89. Gasteiger, J.; Weißenberger, S.; Günnemann, S. Diffusion Improves Graph Learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*. Available online: https://proceedings.neurips.cc/paper_files/paper/2019/file/23c894276a2c5a16470e6a31f4618d73-Paper.pdf (accessed on 6 January 2026).
90. Vignac, C.; Krawczuk, I.; Siraudin, A.; Wang, B.; Cevher, V.; Frossard, P. DiGress: Discrete Denoising Diffusion for Graph Generation. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
91. Ektefaie, Y.; Dasoulas, G.; Noori, A.; Farhat, M.; Zitnik, M. Multimodal Learning with Graphs. *Nat. Mach. Intell.* **2023**, *5*, 340–350.
92. Aggarwal, S.; Kumar, N. Path Planning Techniques for Unmanned Aerial Vehicles: A Review, Solutions, and Challenges. *Comput. Commun.* **2020**, *149*, 270–299. <https://doi.org/10.1016/j.comcom.2019.10.014>.
93. Debnath, D.; Vanegas, F.; Sandino, J.; Hawary, A.F.; Gonzalez, F. A Review of UAV Path-Planning Algorithms and Obstacle Avoidance Methods for Remote Sensing Applications. *Remote Sens.* **2024**, *16*, 4019. <https://doi.org/10.3390/rs16214019>.
94. Van Den Berg, J.; Lin, M.; Manocha, D. Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation. In 2008 IEEE International Conference on Robotics and Automation; IEEE: New York, NY, USA, 2008; pp. 1928–1935. <https://doi.org/10.1109/ROBOT.2008.4543489>.
95. Li, B.; Fei, Z.; Zhang, Y. UAV Communications for 5G and Beyond: Recent Advances and Future Trends. *IEEE Internet Things J.* **2019**, *6*, 2241–2263. <https://doi.org/10.1109/JIOT.2018.2887086>.
96. Liao, X.; Wang, Y.; Zhu, X.; Lin, C.; Han, Y.; Li, Y. Graph Neural Network Assisted Spectrum Resource Optimisation for UAV Swarm. *IET Commun.* **2025**, *19*, e70078. <https://doi.org/10.1049/cmu2.70078>.
97. Marwani, M.; Kaddoum, G. Graph Neural Networks Approach for Joint Wireless Power Control and Spectrum Allocation. *IEEE Trans. Mach. Learn. Commun. Netw.* **2024**, *2*, 717–732. <https://doi.org/10.1109/TMLCN.2024.3408723>.
98. Li, S.; Yeung, D.Y. Visual Object Tracking for Unmanned Aerial Vehicles: A Benchmark and New Motion Models. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
99. Xiao, C.; Cao, Q.; Zhong, Y.; Lan, L.; Zhang, X.; Cai, H.; Luo, Z. Enhancing Online UAV Multi-Object Tracking with Temporal Context and Spatial Topological Relationships. *Drones* **2023**, *7*, 389. <https://doi.org/10.3390/drones7060389>.
100. Xue, Y.; Jin, G.; Shen, T.; Tan, L.; Wang, N.; Gao, J.; Wang, L. SmallTrack: Wavelet Pooling and Graph Enhanced Classification for UAV Small Object Tracking. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5618815. <https://doi.org/10.1109/TGRS.2023.3305728>.
101. Zhang, B.; Hu, M.; Li, Z.; Fei, W.; Yu, Z.; Qi, N.; Wang, S. Graph-Based Multi-Agent Reinforcement Learning for Collaborative Search and Tracking of Multiple UAVs. *Chin. J. Aeronaut.* **2025**, *38*, 103214.
102. Campion, M.; Ranganathan, P.; Faruque, S. UAV Swarm Communication and Control Architectures: A Review. *J. Unmanned Veh. Syst.* **2018**, *7*, 93–106.
103. Zhang, X.; Liu, J. Research on UAV Swarm Network Modeling and Resilience Assessment Methods. *Sensors* **2024**, *24*, 11. <https://doi.org/10.3390/s24010011>.
104. Tang, X.; Li, X.; Yu, R.; Wu, Y.; Ye, J.; Tang, F.; Chen, Q. Digital-Twin-Assisted Task Assignment in Multi-UAV Systems: A Deep Reinforcement Learning Approach. *IEEE Internet Things J.* **2023**, *10*, 15362–15375. <https://doi.org/10.1109/JIOT.2023.3263574>.
105. Peng, J.; Viswanath, H.; Bera, A. Graph-Based Decentralized Task Allocation for Multi-Robot Target Localization. *IEEE Robot. Autom. Lett.* **2024**, *9*, 10676–10683. <https://doi.org/10.1109/LRA.2024.3475013>.
106. Ma, Z.; Xiong, J.; Gong, H.; Wang, X. Mission Planning of UAVs and CAVs Based on Graph Neural Network Transformer Model. *IEEE Internet Things J.* **2024**, *11*, 40532–40546. <https://doi.org/10.1109/JIOT.2024.3451248>.
107. Gerkey, B.P.; Matarić, M.J. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *Int. J. Robot. Res.* **2004**, *23*, 939–954.
108. Zhang, A.; Zhang, B.; Bi, W.; Huang, Z.; Yang, M. Multi-UAV Task Allocation Based on GCN-Inspired Binary Stochastic L-BFGS. *Comput. Commun.* **2023**, *212*, 198–211.
109. Van Der Bergh, B.; Chiumento, A.; Pollin, S. LTE in the Sky: Trading off Propagation Benefits with Interference Costs for Aerial Nodes. *IEEE Commun. Mag.* **2016**, *54*, 44–50. <https://doi.org/10.1109/MCOM.2016.7470934>.
110. Wei, X.; Cui, W.; Huang, X.; Yang, L.; Geng, X.; Tao, Z.; Zhai, Y. Hierarchical RNNs with Graph Policy and Attention for Drone Swarm. *J. Comput. Des. Eng.* **2024**, *11*, 314–326. <https://doi.org/10.1093/jcde/qwae020>.
111. Wang, S.; Eravci, B.; Guliyev, R.; Ferhatosmanoglu, H. Low-Bit Quantization for Deep Graph Neural Networks with Smoothness-Aware Message Propagation. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management; Association for Computing Machinery: New York, NY, USA, 2023; pp. 2626–2636. <https://doi.org/10.1145/3583780.3614955>.
112. Huang, S.; Zhang, H.; Huang, Z. E2CoPre: Energy Efficient and Cooperative Collision Avoidance for UAV Swarms with Trajectory Prediction. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 6951–6963. <https://doi.org/10.1109/TITS.2023.3342161>.
113. Di Franco, C.; Buttazzo, G. Energy-Aware Coverage Path Planning of UAVs. In 2015 IEEE International Conference on Autonomous Robot Systems and Competitions; IEEE: New York, NY, USA, 2015; pp. 111–117. <https://doi.org/10.1109/ICARSC.2015.17>.
114. Tang, J.; Wu, G.; Jalalzai, M.M.; Wang, L.; Zhang, B.; Zhou, Y. Energy-Optimal DNN Model Placement in UAV-Enabled Edge Computing Networks. *Digit. Commun. Netw.* **2024**, *10*, 827–836. <https://doi.org/10.1016/j.dcan.2023.11.002>.

115. Yang, X.; Yang, X.; Yu, C.; Chen, J.; Ding, W.; Yang, H.; Wang, Y. MASP: Scalable Graph-Based Planning Towards Multi-UAV Navigation. *Drones* **2025**, *9*, 463. <https://doi.org/10.3390/drones9070463>.
116. Ivić, S.; Crnković, B.; Grbčić, L.; Matleković, L. Multi-UAV Trajectory Planning for 3D Visual Inspection of Complex Structures. *Autom. Constr.* **2023**, *147*, 104709. <https://doi.org/10.1016/j.autcon.2022.104709>.
117. Feng, S.; Zeng, L.; Liu, J.; Yang, Y.; Song, W. Multi-UAVs Collaborative Path Planning in the Cramped Environment. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 529–538. <https://doi.org/10.1109/JAS.2023.123945>.
118. Marino, A.; Pacchierotti, C.; Giordano, P.R. Multi-UAVs End-to-End Distributed Trajectory Generation over Point Cloud Data. *IEEE Robot. Autom. Lett.* **2024**, *9*, 7629–7636. <https://doi.org/10.1109/LRA.2024.3426374>.
119. Ratnabala, L.; Fedoseev, A.; Peter, R.; Tsetserukou, D. MAGNNET: Multi-Agent Graph Neural Network-Based Efficient Task Allocation for Autonomous Vehicles with Deep Reinforcement Learning. In Proceedings of the 2025 IEEE Intelligent Vehicles Symposium (IV), Cluj-Napoca, Romania, 22–25 June 2025; pp. 970–975. <https://doi.org/10.1109/IV64158.2025.11097641>.
120. Liu, W.H.; Zheng, X.; Deng, Z.H. Dynamic Collision Avoidance for Cooperative Fixed-Wing UAV Swarm Based on Normalized Artificial Potential Field Optimization. *J. Cent. South Univ.* **2021**, *28*, 3159–3172.
121. Arshid, K.; Krayani, A.; Marcenaro, L.; Gomez, D.M.; Regazzoni, C. Toward Autonomous UAV Swarm Navigation: A Review of Trajectory Design Paradigms. *Sensors* **2025**, *25*, 5877.
122. Cheng, Z.; Yang, J.; Sun, J.; Zhao, L. Trajectory Planning of Unmanned Aerial Vehicles in Complex Environments Based on Intelligent Algorithm. *Drones* **2025**, *9*, 468.
123. Ji, Y.; Sun, R.; Wang, Y.; Zhu, Z.; Liu, Z. Dynamic Path Planning for Unmanned Autonomous Vehicles Based on CAS-UNet and Graph Neural Networks. *Sensors* **2025**, *25*, 4283.
124. Hou, Y.; Zhao, J.; Zhang, R.; Cheng, X.; Yang, L. UAV Swarm Cooperative Target Search: A Multi-Agent Reinforcement Learning Approach. *IEEE Trans. Intell. Veh.* **2024**, *9*, 568–578. <https://doi.org/10.1109/TIV.2023.3316196>.
125. Manfreda, S.; McCabe, M.F.; Miller, P.E.; Lucas, R.; Pajuelo Madrigal, V.; Mallinis, G.; Ben Dor, E.; Helman, D.; Estes, L.; Ciraolo, G.; et al. On the Use of Unmanned Aerial Systems for Environmental Monitoring. *Remote Sens.* **2018**, *10*, 641.
126. Xu, Y.; Zha, J.; Hong, W.; Yi, X.; Chen, G.; Zheng, J.; Hsia, C.-C.; Chen, X. Scalable UAV Multi-Hop Networking via Multi-Agent Reinforcement Learning with Large Language Models. *IEEE Trans. Mobile Comput.* **2026**, *25*, 11173–11190. <https://doi.org/10.1109/TMC.2026.3669346>.
127. Wang, Y.; Liao, X.; Ye, G.; Zhu, X. Knowledge Graph Completion Assisted Graph Neural Network for Spectrum Resource Optimization of UAV Swarm. *J. Phys. Conf. Ser.* **2024**, *2717*, 012012.
128. Zhang, Y.; Xiu, S.; Cai, Y.; Ren, P. Scheduling of Graph Neural Network and Markov Based UAV Mobile Edge Computing Networks. *Phys. Commun.* **2023**, *60*, 102160.
129. Jiang, W.; Han, H.; Zhang, Y.; Wang, J.A.; He, M.; Gu, W.; Mu, J.; Cheng, X. Graph Neural Networks for Routing Optimization: Challenges and Opportunities. *Sustainability* **2024**, *16*, 9239.
130. Li, P.; Wang, L.; Wu, W.; Zhou, F.; Wang, B.; Wu, Q. Graph Neural Network-Based Scheduling for Multi-UAV-Enabled Communications in D2D Networks. *Digit. Commun. Netw.* **2024**, *10*, 45–52.
131. Wang, Q.; Lu, Y.; Chen, W.; Ai, B.; Zhong, Z.; Niyato, D. GNN-Enabled Optimization of Placement and Transmission Design for UAV Communications. *IEEE Trans. Veh. Technol.* **2025**, *74*, 6656–6661. <https://doi.org/10.1109/TVT.2024.3514860>.
132. An, Y.; Liu, A.; Liu, H.; Geng, L. Multidimensional Trajectory Prediction of UAV Swarms Based on Dynamic Graph Neural Network. *IEEE Access* **2024**, *12*, 57033–57042. <https://doi.org/10.1109/ACCESS.2024.3391374>.
133. Huo, Y.; Zhang, H.; Tian, Y.; Wang, Z.; Wu, J.; Yao, X. A Spatiotemporal Graph Neural Network with Graph Adaptive and Attention Mechanisms for Traffic Flow Prediction. *Electronics* **2024**, *13*, 212. <https://doi.org/10.3390/electronics13010212>.
134. Xue, J.; Tan, R.; Ma, J.; Ukkusuri, S.V. Data Science in Transportation Networks with Graph Neural Networks: A Review and Outlook. *Data Sci. Transp.* **2025**, *7*, 10.
135. Vrahatis, A.G.; Lazaros, K.; Kotsiantis, S. Graph Attention Networks: A Comprehensive Review of Methods and Applications. *Future Internet* **2024**, *16*, 318. <https://doi.org/10.3390/fi16090318>.
136. Pamuklu, T.; Syed, A.; Kennedy, W.S.; Erol-Kantarci, M. Heterogeneous GNN-RL-Based Task Offloading for UAV-Aided Smart Agriculture. *IEEE Netw. Lett.* **2023**, *5*, 213–217. <https://doi.org/10.1109/LNET.2023.3283936>.
137. Wang, Y.; Wu, J.; Hua, X.; Liu, C.H.; Li, G.; Zhao, J.; Yuan, Y.; Wang, G. Air-Ground Spatial Crowdsourcing with UAV Carriers by Geometric Graph Convolutional Multi-Agent Deep Reinforcement Learning. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*; IEEE: New York, NY, USA, 2023; pp. 1790–1802. <https://doi.org/10.1109/ICDE55515.2023.00140>.
138. Liu, Z.; Zhang, J.; Shi, E.; Liu, Z.; Niyato, D.; Ai, B.; Shen, X. Graph Neural Network Meets Multi-Agent Reinforcement Learning: Fundamentals, Applications, and Future Directions. *IEEE Wirel. Commun.* **2024**, *31*, 39–47. <https://doi.org/10.1109/MWC.015.2300595>.
139. Ekechi, C.C.; Elfouly, T.; Alouani, A.; Khattab, T. A Survey on UAV Control with Multi-Agent Reinforcement Learning. *Drones* **2025**, *9*, 484. <https://doi.org/10.3390/drones9070484>.

140. Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624.
141. Lin, Z.; Li, C.; Miao, Y.; Liu, Y.; Xu, Y. PaGraph: Scaling GNN Training on Large Graphs via Computation-Aware Caching. *Proc. ACM Symp. Cloud Comput.* **2020**, 401–415. <https://doi.org/10.1145/3419111.3421281>.
142. Wan, X.; Xu, K.; Liao, X.; Jin, Y.; Chen, K.; Jin, X. Scalable and Efficient Full-Graph GNN Training for Large Graphs. *Proc. ACM Manag. Data* **2023**, *1*, 143. <https://doi.org/10.1145/3589288>.
143. Huang, W.; Zhang, T.; Rong, Y.; Huang, J. Adaptive Sampling towards Fast Graph Representation Learning. *Adv. Neural Inf. Process. Syst.* **2018**, *31*. Available online: https://proceedings.neurips.cc/paper_files/paper/2018/file/01eee509ee2f68dc6014898c309e86bf-Paper.pdf (accessed on 8 January 2026).
144. Liu, X.; Yan, M.; Deng, L.; Li, G.; Ye, X.; Fan, D. Sampling Methods for Efficient Training of Graph Convolutional Networks: A Survey. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 205–234.
145. Hu, K.; Pan, H.; Han, C.; Sun, J.; An, D.; Li, S. Graph Neural Network-Enhanced Multi-Agent Reinforcement Learning for Intelligent UAV Confrontation. *Aerospace* **2025**, *12*, 687.
146. Ren, J.; Xu, Y.; Li, Z.; Hong, C.; Zhang, X.P.; Chen, X. Scheduling UAV Swarm with Attention-Based Graph Reinforcement Learning for Ground-to-Air Heterogeneous Data Communication. In *Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 670–675.
147. Tang, X.; Zhao, K.; Shen, C.; Du, Q.; Wang, Y.; Niyato, D.; Han, Z. Deep Graph Reinforcement Learning for UAV-Enabled Multi-User Secure Communications. *IEEE Trans. Mob. Comput.* **2025**, *24*, 8780–8793. <https://doi.org/10.1109/TMC.2025.3558790>.
148. Yu, J.; Zhang, Y.; Sun, C. Multi-UAV Dynamic Task Assignment Based on Event-Triggered Graph Reinforcement Learning under Weak Communication. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 19106–19121. <https://doi.org/10.1109/TASE.2025.3593125>.
149. Kaler, T.; Iliopoulos, A.; Murzynowski, P.; Schardl, T.; Leiserson, C.E.; Chen, J. Communication-Efficient Graph Neural Networks with Probabilistic Neighborhood Expansion Analysis and Caching. *Proc. Mach. Learn. Syst.* **2023**, *5*, 477–494.
150. Blumenkamp, J.; Morad, S.; Gielis, J.; Li, Q.; Prorok, A. A Framework for Real-World Multi-Robot Systems Running Decentralized GNN-Based Policies. In *Proceedings of the 2022 International Conference on Robotics and Automation (ICRA)*, Philadelphia, PA, USA, 23–27 May 2022; pp. 8772–8778. <https://doi.org/10.1109/ICRA46639.2022.9811744>.
151. Yao, J.; Li, W.; Zheng, Z.; Jia, X. CLGNN: UAV Fault Diagnosis via Causal Learning and Graph Neural Network. In *2024 International Joint Conference on Neural Networks (IJCNN)*; IEEE: New York, NY, USA, 2024; pp. 1–8. <https://doi.org/10.1109/IJCNN60899.2024.10651352>.
152. Wang, S.; Liu, Z.; Jia, Z.; Tang, Y.; Zhi, G.; Wang, X. Fault Detection for UAVs with Spatial-Temporal Learning on Multivariate Flight Data. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 2529517. <https://doi.org/10.1109/TIM.2024.3440387>.
153. He, K.; Yu, D.; Wang, D.; Chai, M.; Lei, S.; Zhou, C. Graph Attention Network-Based Fault Detection for UAVs with Multivariate Time Series Flight Data. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3530213. <https://doi.org/10.1109/TIM.2022.3219489>.
154. Abro, G.E.M.; Abdallah, A.M. Graph Attention Networks for Anomalous Drone Detection: RSSI-Based Approach with Real-World Validation. *Expert Syst. Appl.* **2025**, *273*, 126913.
155. Peng, Y.; Wu, J.; Duan, T.; Liu, Y.; Zhang, Z.; Zhang, J. Prioritized Recovery Strategy for Robust UAV Swarm Communication via Graph Reinforcement Learning. *IEEE Internet Things J.* **2025**, *12*, 23891–23904. <https://doi.org/10.1109/JIOT.2025.3553919>.
156. Sankar, A.; Wu, Y.; Gou, L.; Zhang, W.; Yang, H. DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 519–527.
157. Sun, J.; Su, L.; Shi, Z.; Shen, W.; Wang, Z.; Wang, L.; Zhang, J.; Li, Y.; Yu, W.; Zhou, J.; Wu, F. Legion: Automatically Pushing the Envelope of Multi-GPU System for Billion-Scale GNN Training. In *Proceedings of the 2023 USENIX Annual Technical Conference*, Boston, MA, USA, 10–12 July 2023; pp. 165–179.
158. Chen, W.; He, S.; Qu, H.; Zhang, X. LeapGNN: Accelerating Distributed GNN Training Leveraging Feature-Centric Model Migration. In *Proceedings of the 23rd USENIX Conference on File and Storage Technologies*, Santa Clara, CA, USA, 25–27 February 2025; pp. 255–270.
159. Song, C.; Lin, Y.; Guo, S.; Wan, H. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Palo Alto, CA, USA, 2020; Volume 34, pp. 914–921. <https://doi.org/10.1609/aaai.v34i01.5438>.
160. Zhang, C.; Patras, P.; Haddadi, H. Deep Learning in Mobile and Wireless Networking: A Survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2224–2287. <https://doi.org/10.1109/COMST.2019.2904897>.
161. Williams, R.J. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.* **1992**, *8*, 229–256.
162. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2018; pp. 1861–1870.

163. Pathak, D.; Agrawal, P.; Efros, A.A.; Darrell, T. Curiosity-Driven Exploration by Self-Supervised Prediction. In *International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2017; pp. 2778–2787.
164. Bengio, Y.; Louradour, J.; Collobert, R.; Weston, J. Curriculum Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*; Association for Computing Machinery: New York, NY, USA, 2009; pp. 41–48.
165. Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; Tang, J. Self-Supervised Learning: Generative or Contrastive. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 857–876. <https://doi.org/10.1109/TKDE.2021.3090866>.
166. Abreha, H.G.; Hayajneh, M.; Serhani, M.A. Federated Learning in Edge Computing: A Systematic Survey. *Sensors* **2022**, *22*, 450. <https://doi.org/10.3390/s22020450>.
167. Aygün, O.; Kazemi, M.; Gündüz, D.; Duman, T.M. Over-the-Air Federated Edge Learning with Hierarchical Clustering. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 17856–17871. <https://doi.org/10.1109/TWC.2024.3457591>.
168. Xiao, B.; Yu, X.; Ni, W.; Wang, X.; Poor, H.V. Over-the-Air Federated Learning: Status Quo, Open Challenges, and Future Directions. *Fundam. Res.* **2024**, *5*, 1710–1724.
169. Triyanto, D.; Mustika, I.W.; Widyawan. Computation Offloading and Resource Allocation for Energy-Harvested MEC in an Ultra-Dense Network. *Sensors* **2025**, *25*, 1722.
170. Peng, C.; Wang, Q.; Zhang, D. Efficient Dynamic Task Offloading and Resource Allocation in UAV-Assisted MEC for Large Sport Event. *Sci. Rep.* **2025**, *15*, 11828.
171. Wang, S.; Song, X.; Song, T.; Yang, Y. Fairness-Aware Computation Offloading with Trajectory Optimization and Phase-Shift Design in RIS-Assisted Multi-UAV MEC Network. *IEEE Internet Things J.* **2024**, *11*, 20547–20561. <https://doi.org/10.1109/JIOT.2024.3371395>.
172. Liu, J.; Hu, H.; Bai, X.; Li, G.; Zhang, X.; Zhou, H.; Li, H.; Liu, J. Multi-UAV-Assisted Task Offloading and Trajectory Optimization for Edge Computing via NOMA. *Sensors* **2025**, *25*, 4965.
173. Yu, P.; Zhang, J.; Fang, H.; Li, W.; Feng, L.; Zhou, F.; Xiao, P.; Guo, S. Digital Twin Driven Service Self-Healing with Graph Neural Networks in 6G Edge Networks. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 3607–3623. <https://doi.org/10.1109/JSAC.2023.3310063>.
174. Ngo, D.T.; Aouedi, O.; Piamrat, K.; Hassan, T.; Raipin-Parvédy, P. Empowering Digital Twin for Future Networks with Graph Neural Networks: Overview, Enabling Technologies, Challenges, and Opportunities. *Future Internet* **2023**, *15*, 377.
175. Yin, P.; Liang, W.; Wen, J.; Kang, J.; Chen, J.; Niyato, D. Multi-Agent Deep Reinforcement Learning for Multi-Objective Twin Migration Routing with Workload Prediction in 6G-Enabled Internet of Vehicles. *IEEE Trans. Mobile Comput.* **2026**, *25*, 12222–12236. <https://doi.org/10.1109/TMC.2026.3672525>.
176. Jiang, X.; Sheng, M.; Zhao, N.; Xing, C.; Lu, W.; Wang, X. Green UAV Communications for 6G: A Survey. *Chin. J. Aeronaut.* **2022**, *35*, 19–34. <https://doi.org/10.1016/j.cja.2021.04.025>.
177. Othman, W.M.; Ateya, A.A.; Nasr, M.E.; Muthanna, A.; ElAffendi, M.; Koucheryavy, A.; Hamdi, A.A. Key Enabling Technologies for 6G: The Role of UAVs, Terahertz Communication, and Intelligent Reconfigurable Surfaces in Shaping the Future of Wireless Networks. *J. Sens. Actuator Netw.* **2025**, *14*, 30. <https://doi.org/10.3390/jsan14020030>.
178. Mozaffari, M.; Lin, X.; Hayes, S. Toward 6G with Connected Sky: UAVs and Beyond. *IEEE Commun. Mag.* **2021**, *59*, 74–80. <https://doi.org/10.1109/MCOM.005.2100142>.
179. Mahbub, M.; Saym, M.M.; Jahan, S.; Paul, A.K.; Vahid, A.; Hosseinalipour, S.; Barua, B.; Yeh, H.G.; Shubair, R.M.; Taleb, T.; Kaushik, A. UAV-Assisted Wireless Communications in the 6G-and-Beyond Era: An Extensive Survey on Characteristics, Standardization and Regulations, Enabling Technologies, Challenges, and Future Directions. *Veh. Commun.* **2025**, *56*, 100977. <https://doi.org/10.1016/j.vehcom.2025.100977>.
180. Akbar, M.S.; Hussain, Z.; Ikram, M.; Sheng, Q.Z.; Mukhopadhyay, S.C. On Challenges of Sixth-Generation (6G) Wireless Networks: A Comprehensive Survey of Requirements, Applications, and Security Issues. *J. Netw. Comput. Appl.* **2025**, *233*, 104040.
181. Giordani, M.; Zorzi, M. Non-Terrestrial Networks in the 6G Era: Challenges and Opportunities. *IEEE Netw.* **2021**, *35*, 244–251. <https://doi.org/10.1109/MNET.011.2000493>.
182. Mustari, N.; Karabulut, M.A.; Shah, A.S.; Tureli, U. Cooperative THz Communication for UAVs in 6G and Beyond. *Green Energy Intell. Transp.* **2024**, *3*, 100131.
183. Khan, M.A.; Kumar, N.; Mohsan, S.A.; Khan, W.U.; Nasralla, M.M.; Alsharif, M.H.; Żywiołek, J.; Ullah, I. Swarm of UAVs for Network Management in 6G: A Technical Review. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 741–761. <https://doi.org/10.1109/TNSM.2022.3213370>.
184. Moorthy, S.K.; Jagannath, J. Survey of Graph Neural Network for Internet of Things and NextG Networks. *IEEE Commun. Surv. Tutor.* **2026**, *28*, 1803–1844. <https://doi.org/10.1109/COMST.2025.3625399>.
185. Njanko, A.; Rawat, D.B.; Kamhoua, C.; Hemida, A. An Unsupervised Graph Neural Network Approach to Deceive UAV Network Reconnaissance Attack. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*; IEEE: New York, NY, USA, 2024; pp. 1–6. <https://doi.org/10.1109/INFOCOMWKSHPS61880.2024.10620885>.

186. Krishnan, S.; Park, J.; Sherman, G.; Campbell, B.; Choi, J. Graph Koopman Autoencoder for Predictive Covert Communication against UAV Surveillance. In *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*; IEEE: New York, NY, USA, 2024; pp. 1–5. <https://doi.org/10.1109/VTC2024-Spring62846.2024.10683140>.
187. Tang, X.; Chen, Q.; Li, F.; Gong, Y.; Liu, Y.; Tian, W.; Qin, S.; Li, X. Topology Generation of UAV Covert Communication Networks: A Graph Diffusion Approach with Incentive Mechanism. In *2025 4th International Conference on Artificial Intelligence, Internet of Things and Cloud Computing Technology (AIoTC)*; IEEE: New York, NY, USA, 2025; pp. 627–632. <https://doi.org/10.1109/AIoTC66747.2025.11198628>.
188. Alshehri, S.M.; Sharaf, S.A.; Molla, R.A. Systematic Review of Graph Neural Network for Malicious Attack Detection. *Information* **2025**, *16*, 470.
189. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.; Poor, H.V. Federated Learning with Differential Privacy: Algorithms and Performance Analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469.
190. Yavuz, M.; Kaya, M. A Method for Anomaly Detection of Unmanned Aerial Vehicles (UAVs). In *2024 Mediterranean Smart Cities Conference (MSCC)*; IEEE: New York, NY, USA, 2024; pp. 1–6. <https://doi.org/10.1109/MSCC62288.2024.10697079>.
191. Zhang, Y.; Sun, Y.; Wang, R.; Tai, Y. Research on Situation Awareness Strategy for Drone Swarm Defense. *J. Comput. Methods Sci. Eng.* **2025**, *25*, 1–7. <https://doi.org/10.1177/14727978251361824>.
192. Feng, C.; Fan, J.; Liu, Z.; Jin, G.; Chen, S. Unmanned Aerial Vehicle Anomaly Detection Based on Causality-Enhanced Graph Neural Networks. *Drones* **2025**, *9*, 408. <https://doi.org/10.3390/drones9060408>.
193. Gao, Y.; Yang, H.; Zhang, P.; Zhou, C.; Hu, Y. Graph Neural Architecture Search. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence; IJCAI: Yokohama, Japan, 2020; Volume 20*, pp. 1403–1409. <https://doi.org/10.24963/ijcai.2020/195>.
194. Zhou, K.; Huang, X.; Song, Q.; Chen, R.; Hu, X. Auto-GNN: Neural Architecture Search of Graph Neural Networks. *Front. Big Data* **2022**, *5*, 1029307. <https://doi.org/10.3389/fdata.2022.1029307>.
195. Erdelj, M.; Król, M.; Natalizio, E. Wireless Sensor Networks and Multi-UAV Systems for Natural Disaster Management. *Comput. Netw.* **2017**, *124*, 72–86.
196. Huo, L.; Wang, C.; Han, Y. Autonomous Air Combat decision-making via Graph Neural Networks and Reinforcement Learning. *Sci. Rep.* **2025**, *15*, 16169. <https://doi.org/10.1038/s41598-025-00463-y>.
197. Hazarika, B.; Singh, P.; Singh, K.; Dobre, O.A.; Duong, T.Q. Dynamic UAV Swarm Control in Disaster Recovery via GenAI-Based Graph Reinforcement Learning. In *ICC 2025-IEEE International Conference on Communications*; IEEE: New York, NY, USA, 2025; pp. 1–6. <https://doi.org/10.1109/ICC52391.2025.11161965>.
198. Guo, T.; Mei, Y.; Du, W.; Lv, Y.; Li, Y.; Song, T. Emergency Scheduling of Aerial Vehicles via Graph Neural Neighborhood Search. *IEEE Trans. Artif. Intell.* **2025**, *6*, 1808–1822. <https://doi.org/10.1109/TAI.2025.3528381>.
199. Tolstaya, E.; Paulos, J.; Kumar, V.; Ribeiro, A. Multi-Robot Coverage and Exploration Using Spatial Graph Neural Networks. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: New York, NY, USA, 2021; pp. 8944–8950.
200. Dorling, K.; Heinrichs, J.; Messier, G.G.; Magierowski, S. Vehicle Routing Problems for Drone Delivery. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *47*, 70–85. <https://doi.org/10.1109/TSMC.2016.2582745>.
201. Seo, J.; Duque, L.; Wacker, J. Drone-Enabled Bridge Inspection Methodology and Application. *Autom. Constr.* **2018**, *94*, 112–126.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.