

Article

Adaptive k -Truss-Constrained Agentic AI Framework for Resilient Multi-Agent UAV Swarm Coordination in Dynamic Disaster Environments

Hedi Hamdi ^{1,*}  and Nabil Almashfi ² ¹ Department of Computer Science, Jouf University, Sakaka 72388, Saudi Arabia² Department of Software Engineering, Jouf University, Sakaka 72388, Saudi Arabia; nalmashfi@ju.edu.sa

* Correspondence: hhamdi@ju.edu.sa

Abstract

Coordinating multi-agent unmanned aerial vehicle (UAV) swarms is challenging in disaster scenarios where communications are dynamic, unreliable, and subject to UAV losses. While distributed artificial intelligence has enabled unprecedented levels of autonomous multi-agent coordination, most methods implicitly take communication topology as a given, not accounting for its limited maintenance in such scenarios. As a result, communication fragmentation can undermine autonomous mission progress during highly dynamic communications conditions. This paper proposes the Adaptive k -Truss-Constrained Agentic AI Framework (ATAC), an AI-based decentralized coordination framework for multi-agent UAV systems, which explicitly factors in graph-theoretic structural considerations during decision-making. The swarm is modeled as a graph, where an adaptive k -truss backbone is maintained during dynamic communication conditions to preserve triangle-based redundancy. Each agent acts as a graph-aware AI entity which bases its decentralized decisions on local information and descriptors of the communication backbone. A closed-loop evolutionary process is used to rebuild the backbone after significant communication link losses while UAVs make mission progress decisions based on information from the current backbone, enabling continuous adaption of the swarm structure to the communication state. The efficacy of the proposed framework is demonstrated through a comprehensive simulation campaign which includes communication link losses, UAV failures, adaptive truss selection, ablation studies, reward sensitivity analysis, and computational performance assessments. ATAC is compared to alternative graph-aware coordination approaches, showing consistent improvements in maintaining communication, preserving backbone structure, enabling triangle-based connectivity, and overall structural recovery while still achieving high-levels of mission progress during dynamic disaster response scenarios. The results highlight the effectiveness of explicitly tying AI-driven decentralized decision-making to maintenance of a graph-theoretic backbone structure for resilient UAV swarm coordination.

Keywords: agentic AI; artificial intelligence; graph-aware coordination; adaptive k -truss; decentralized decision-making; UAV swarms; disaster response; network resilience



Academic Editors: Hocine Cherifi,
Christophe Cruz and Maria Alice
Bertolim Nicoleau

Received: 3 July 2026

Revised: 28 August 2026

Accepted: 31 August 2026

Published: 6 September 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.

1. Introduction

Natural disasters (e.g., earthquakes, floods, wildfire) are large-scale, dynamic, uncertain, partially observable, and disruptive events. A disaster response can be viewed from a systems and control point of view as a dynamic stochastic networked system in which

sensing, communication and decision-making operations need to be executed on the system in a timely and reliable manner [1,2]. In such settings, the capability of quickly gathering information, staying connected, and ensuring a coordinated action of distributed agents is paramount to reduce human and economic damage. Recent advances in autonomous agents and agentic artificial intelligence have expanded the capabilities of distributed decision-making systems, enabling perception, reasoning, planning, and adaptive action selection in complex dynamic environments [3,4].

Unmanned aerial vehicle (UAV) swarms have been recognized as a key enabling technology in disaster management, emergency communications, and large-scale environmental monitoring applications [5,6]. In the environment where it is impossible or unsafe to go to with human responders, UAV swarms can be used to support distributed sensing, cooperative exploration, and real-time (or near-real time) communication services [1,7]. Swarm systems can become robust by utilizing redundancy and parallelism by using decentralized systems. But the benefits are high complexity in coordination and communication, particularly in the case of dynamic topologies where the topology is constantly changing and the reliability is not guaranteed.

Disaster scenarios can be formally modeled as a decentralized multi-agent decision problem over a time evolving interaction graph $G_t = (V_t, E_t)$, the nodes in which correspond to the agents and the edges to the communication links. When the agents interact with each other and when they must adapt to environmental changes, each agent has to make decisions based on incomplete information. The goal is to optimize a global reward function representing mission objectives (e.g., coverage, search efficiency, resource allocation). Non-stationarity and stochasticity of the environment pose many difficulties in achieving convergence, stability, and scalability among decentralized policies [8,9].

In the past few years, decentralized coordination problems have been tackled using multi-agent reinforcement learning (MARL). In MARL, policies are learned from interacting with the environment, allowing for adaptive behavior in uncertain environments. We follow the idea of *agentic AI* which are autonomous agents with perception, reasoning and action selection abilities within an shared environment. Formally, the decentralized decision policy of each agent i is

$$\pi_i(a_i \mid o_i, G_t),$$

where o_i denotes the local observation of agent i and G_t represents the current communication graph. In the present implementation, this policy is not learned through neural-network training or policy-gradient optimization; rather, it is a graph-aware decision policy that uses local observations together with graph-derived structural information for decentralized action selection. This formulation enables a team of agents to work autonomously and focus on local tasks to meet global goals [10–12].

Although decentralized multi-agent coordination has seen tremendous success, most such methods have focused on optimizing task level objectives, such as coverage, routing, or energy consumption while only indirectly accounting for communication topologies as implicit structures. In particular, it is common for communication to only be considered as a binary function of distance with no notion of maintenance or reliability. Connectivity is typically assumed to be perfect or implicitly controlled through positioning, which can lead to policies that perform poorly in the presence of realistic communication disruptions. In a theoretical sense, this restriction may be explained by analysing the interaction graph G_t . It is important to note that, connectivity only really captures the extent to which G_t is connected, and this is not sufficient for robust coordination—redundancy or fault tolerance is also necessary. More specifically, if the network is fragmented due to the loss of a few nodes or links, there will be a substantial disruption of coordination and information

transmission. Therefore a more developed idea of network cohesion is needed to ensure resilience in dynamic environments.

The task of capturing such structural properties requires the development of a principled model based on graphs, which is provided by the models of cohesive subgraphs. In these, the k -truss is a very interesting model as it detects highly connected sub-cliques by triangle support. A k -truss is a maximal subgraph such that all its edges are involved in at least $k - 2$ triangles, which provides multiple alternative paths for local reinforcement of the subgraph [13]. Unlike the k -core decomposition that is based only on the degrees of nodes, k -truss is able of capturing more complex connectivity patterns and offers more guarantees of robustness under edge/node failure [14,15].

Much effort has gone into efficient computation of k -truss subgraphs in large scale graphs. Early efforts featured the scalable decomposition scheme with support-based edge peeling [13] and later, parallelization and data structure optimization [14]. Further extensions are the weighted k -truss models [16] and the truss-based optimization problems [17] as well as applications for community detection. Recently, the dynamic k -truss maintenance for evolving graphs has been investigated, such as the batch update algorithms in [18] and fully dynamic maintenance frameworks in [19] and probabilistic models for uncertain graphs in [20].

But despite these developments, the current approaches to k -truss are mostly for analysing structures and not to be used in real-time decision-making systems. In particular, they do not provide any direct support for the type or functioning of the agents acting on the network and hence do not play a direct role in coordination strategies. In contrast, the policies derived from the AI-based swarm coordination methods are not guaranteed to be robust and do not take into account structural cohesion in the policies' optimization objective.

Although dramatic advances have been made in both graph-theoretic network resilience and AI-enabled decentralized coordination, the two research areas were previously disjointed. Namely, graph-based approaches typically provide rigorous performance guarantees but offer little flexibility in terms of autonomous behaviors, while AI-based methods allow decentralized UAVs to devise their own behaviors but typically do not formally account for network topology structure and its resulting implications on communication performance. The challenge, therefore, is to develop a unified framework that builds upon graph-theoretical structural resilience while also introducing decentralized AI coordination mechanisms, thus enabling both communication robustness and emergent autonomous decision-making capabilities.

The narrow areas mentioned above drive the need for a common framework that can jointly ensure structural resilience and decentralized, yet adaptive control in the case of dynamic UAV swarms. Specifically, it is essential that resilient coordination in disaster-response environments not only involve intelligent decision-making but also continuously maintain reliable communication structures under uncertainty, mobility and link failures.

To mitigate these problems in this paper, a graph-constrained agentic coordination framework is proposed for resilient UAV swarm. The proposed method puts together dynamic graph modeling, adaptive k -truss backbone maintenance and graph-aware decentralized decision-making into a closed loop architecture. The framework proposed here is different in that the topology management and agent coordination issues are explicitly coupled with adaptive multi-agent behavior, unlike current approaches which handle them independently.

The main contributions of this work are listed below.

- We formulate robust UAV swarm coordination as a dynamic graph-constrained AI coordination problem where communication topology and decentralized decisions are mutually adaptive.
- We propose an adaptive backbone maintenance mechanism that ensures dynamicity of k -truss supported communication redundancy.
- We introduce a graph-aware AI coordination policy where each UAV makes decentralized decisions by combining local observations with graph descriptors.
- We develop a closed-loop graph evolution framework involving topology adaption, backbone restructuring and decentralized coordination.
- We evaluate the framework in a comprehensive simulation campaign involving communication outages, UAV failures, adaptive truss selection, reward sensitivity analysis, ablation studies, and scalability benchmarking.

The remainder of this paper is organized as follows: Section 2 reviews the related works. Section 3 introduces preliminaries and the graph-theoretic concepts to be used in this study. Section 4 presents the motivating scenario and the overall problem setting. Section 5 formally defines the resilient coordination problem. Section 6 describes the Adaptive k -Truss-Constrained Agentic AI framework for resilient coordination. Section 7 reports the simulation-based evaluation and performance analysis. Finally, Section 8 concludes this paper and presents suggestions for future work.

2. Related Work

The problem of resilient coordination in multi-agent UAV swarms in uncertain and dynamic environments has gained ever growing interest from various fields of research such as graph-theoretic, distributed systems and artificial intelligence. If viewed from a mathematical perspective, such systems can be described as evolving graphs $G_t = (V_t, E_t)$, where nodes are agents and edges are links of communication or cooperation among agents. The main challenges to achieving robustness in these systems involve keeping them connected, and ensuring that the higher-order attributes of connectivity, such as redundancy, fault tolerance and local cohesion are maintained. Cohesive subgraph models, especially k -truss, have proven to be useful tools that capture triangle-based connectivity and structural resilience [13–16] in this context.

Meanwhile, the progress of multi-agent reinforcement learning (MARL) and distributed AI led to the development of scalable and adaptive coordination strategies for UAV swarms with uncertainty and partial observability [8,9,11,12]. These methods treat each agent as an independent decision maker that is able to learn policies from interacting with the environment and other agents. In spite of the success, most of the existing approaches are only concerned with the communication graph being implicit part of the system without imposing any explicit structural guarantees. This introduces a fundamental discrepancy between graph-theoretic robustness and AI-driven coordination, a gap that remains largely unexplored in the literature.

2.1. k -Truss and Cohesive Subgraph Models

The cohesive subgraph discovery problem is a basic graph mining problem that seeks to find coherent subgraphs that model significant structure or function of a graph. The traditional models, like cliques, give excellent guarantees but are not scalable to large networks in computation time, and relaxed models, like k -core, are scalable but lack precision in structure. The k -truss model is an intermediate option between these extremes, required adding triangle-based edge support and captured the higher-order connectivity without hurting the time efficiency [13].

Following this work, Wang and Cheng [13] proposed efficient algorithms for decomposing a truss to build on their work with the “peeling” method based on support. In fact, Conte et al. [14] later presented scalable and parallel algorithm that significantly boosts performance on large-scale graphs. These works are the foundation for making the k -truss a practical tool to analyze large networks. However, they are mostly concerned with static graphs and ignore the dynamic/real-time aspects.

These have been extended further to make k -truss models more expressive. By adding triangle graph representations for improving computational efficiency, Rezvani and Rezvani [15] improved the computational efficiency, while Zheng et al. [16] added weighted k -truss models for considering the heterogeneity of edge importance. For communication networks, which often have link quality that changes at different times, these extensions are very relevant. However, the above methods are still analytical in nature and not incorporated into the decision-making framework.

In addition to analysis of structures, k -truss has also been applied to the research of the robustness and vulnerability of the network. Zhu et al. [17] explored the k -truss minimization problem and found edges in the graph that are crucial for eroding the graph’s cohesion. The problems highlighted in this line of work are of relevance to resilience as a proxy for k -truss. Such approaches are normally employed offline, and do not tackle the issues of real-time influence and guidance of autonomous agents through the use of such structural properties.

2.2. Dynamic k -Truss and Evolving Graphs

Real networks, such as UAV communication networks, are dynamic in nature, as their topology evolves as result of the mobility of the participating UAVs, environment and failures. This has inspired the creation of adaptive k -truss algorithms that preserve the structure after an update in the graph. Luo et al. [18] presented batch-update approaches to efficiently maintain trusses, which showed that only incremental updates can dramatically save computation cost compared with computing the truss from scratch. This was later extended to fully dynamic graphs, where both the edges are inserted and deleted by Luo et al. [19].

This latter aspect has been pursued in more realistic settings in recent studies. Giving consideration to unreliable or intermittent edges, Sun et al. [20] applied the probabilistic k -truss decomposition to uncertain graph. This is especially useful for UAV networks deployed in disaster scenarios, where communications links are inherently fragile. Distributed implementations of truss maintenance have been explored in other works, allowing scalability of truss maintenance in large and decentralized systems.

Although many advances have been made, dynamic k -truss techniques are still largely concerned with structure correctness, rather than intelligent coordination. They treat the graph as an entity that is evolving outside the control of agents in particular. The topology is, however, directly affected by agents’ decisions in UAV swarms, including agent movement and communication strategy. So, one of the main drawbacks of the current dynamic k -truss methods is that they have not yet been able to be integrated within decision-making processes.

Furthermore, existing dynamic graph algorithms focus on minimizing the computation time and do not align with system-level goals like system resilience, adaptability or mission success. This gap between the aims of the algorithms and the application requirements is not conducive to immediate use in a disaster response context, where timely and adequate coordination is paramount.

2.3. Resilient UAV Swarm Coordination

Maintaining resilient coordination in UAV swarms has emerged as a critical challenge due to the increasing complexity of disaster-response, surveillance, and emergency communication missions. Unlike conventional mobile networks, UAV swarms operate in highly dynamic environments where topology changes continuously as a result of mobility, environmental conditions, communication disruptions, and node failures.

Recent research has investigated a variety of resilience-oriented coordination strategies, including communication-aware trajectory planning, distributed task allocation, adaptive relay deployment, and topology-preserving mobility control. These approaches aim to maintain connectivity and mission continuity despite adverse operating conditions.

For example, Ning et al. [21] studied trajectory optimization under dynamic communication constraints, demonstrating the importance of jointly considering mobility and networking decisions. Similarly, Hazarika et al. [22] proposed intelligent coordination mechanisms for disaster-response UAV swarms operating under uncertain environmental conditions.

Recent graph-aware and transformer-based UAV swarm architectures have also demonstrated improved adaptability and scalability in large-scale multi-agent systems [10,23]. These methods exploit structural information during decision-making and have shown promising results in dynamic and partially observable environments.

Despite these advances, most existing approaches primarily focus on mission optimization and adaptive decision-making. Explicit preservation of higher-order structural cohesion remains largely unexplored. Consequently, many existing UAV coordination frameworks may preserve basic connectivity while failing to maintain robust triangle-supported communication structures capable of tolerating multiple simultaneous failures.

This limitation motivates the integration of graph-theoretic cohesion mechanisms directly into the coordination process, enabling UAV swarms to maintain both operational effectiveness and structural resilience.

2.4. AI-Based Coordination for UAV Swarms

AI-based approaches, particularly multi-agent reinforcement learning, have become the dominant paradigm for UAV swarm coordination. These methods enable decentralized agents to learn adaptive policies through interaction with the environment, making them well-suited for dynamic and uncertain scenarios [8,9]. Applications include routing optimization, task allocation, trajectory planning, and communication management [11,21].

Recent work has incorporated graph-based representations into learning frameworks. In graph-aware reinforcement learning, interactions among agents are modeled explicitly through communication graphs, enabling policies to exploit topological information during decentralized decision-making [22,24]. The emergence of graph foundation models has further demonstrated the potential of exploiting structural priors for scalable learning and decision-making in large dynamic networks [25,26]. By encoding communication relationships into graph representations, these approaches improve information propagation and facilitate more scalable coordination among distributed UAVs operating in dynamic environments. Such methods represent an important step toward integrating graph-theoretic and artificial intelligence within autonomous swarm systems.

In parallel, transformer-based coordination architectures have emerged as promising solutions for modeling long-range dependencies and complex interaction patterns within large-scale UAV swarms. Recent studies have demonstrated that transformer-based mechanisms can improve adaptability, task allocation efficiency, and coordination robustness under dynamic operating conditions [10,23]. These developments highlight the growing convergence between modern AI architectures and network-aware swarm intelligence,

enabling UAV teams to coordinate more effectively in environments characterized by dynamic topologies, partial observability, and uncertain communication conditions.

Despite these advances, a fundamental limitation remains: the graph is typically treated as an informational feature rather than as a constrained structural object. While agents may learn from graph representations, they are generally not required to preserve explicit topological properties such as triangle-supported cohesion or k-truss connectivity. Consequently, communication robustness emerges only indirectly through the learning process and is not explicitly guaranteed. Under severe communication disruptions, learned policies may therefore fail to maintain structurally resilient communication backbones.

Furthermore, many MARL-based coordination strategies exhibit scalability, convergence, and stability challenges in highly dynamic environments, particularly when communication links are intermittent or unreliable. Because communication topology often evolves as a consequence of agent actions, failures in the underlying network may propagate to the decision-making process itself. These limitations are especially problematic in disaster-response scenarios, where maintaining both operational effectiveness and communication resilience is critical. Such observations motivate the need for coordination frameworks that jointly consider adaptive decision-making and explicit structural robustness.

2.5. Comparative Analysis and Research Gap

Although substantial progress has been achieved in graph-theoretic resilience analysis and AI-driven UAV swarm coordination, the two research directions have largely evolved independently.

Graph-theoretic approaches focus on discovering, maintaining, and analyzing structurally cohesive communication patterns. These methods provide strong guarantees regarding robustness, redundancy, and higher-order connectivity, but typically do not influence the behavior of autonomous agents during mission execution.

To facilitate a systematic comparison of representative graph-theoretic approaches, Table 1 summarizes key developments in structural cohesion modeling and dynamic k-truss maintenance.

Table 1. Representative graph-theoretic approaches related to structural cohesion and dynamic k-truss maintenance. ✓: supported, ✗: not supported.

Reference	Dynamic	k-Truss	Weighted	Probabilistic
Wang and Cheng [13]	✗	✓	✗	✗
Conte et al. [14]	✗	✓	✗	✗
Rezvani and Rezvani [15]	✗	✓	✗	✗
Zheng et al. [16]	✗	✓	✓	✗
Zhu et al. [17]	✗	✓	✗	✗
Luo et al. [18]	✓	✓	✗	✗
Luo et al. [19]	✓	✓	✗	✗
Sun et al. [20]	✓	✓	✗	✓
Proposed	✓	✓	✓	✓

As shown in Table 1, substantial progress has been achieved in efficient k-truss computation, weighted cohesion analysis, and dynamic maintenance of evolving graph structures. However, these methods remain primarily analytical and are generally not integrated into autonomous decision-making frameworks. As a result, structural cohesion is treated as an observed property rather than a control objective.

In parallel, significant advances have been achieved in UAV swarm coordination through artificial intelligence, multi-agent learning, and graph-aware decision-making.

Table 2 summarizes representative coordination approaches and highlights their capabilities with respect to resilience and disaster-response applications.

Table 2. Representative UAV swarm coordination approaches and comparison with the proposed framework. ✓: supported; ✗: not supported, Δ: partially supported.

Reference	UAV Swarm	AI-Based	Graph-Aware	Resilience-Oriented	Disaster Scenario
Wang et al. [11]	✓	✓	✗	✗	Δ
Ning et al. [21]	✓	✓	✗	✓	✓
Peng et al. [24]	✓	✓	✓	Δ	Δ
Hazarika et al. [22]	✓	✓	✓	✓	✓
Alam et al. [10]	✓	✓	✓	Δ	Δ
López-Villegas et al. [23]	✓	✓	✓	Δ	✓
Proposed Approach	✓	✓	✓	✓	✓

The comparison presented in Table 2 reveals that recent UAV coordination frameworks increasingly incorporate graph-aware representations and adaptive learning mechanisms. Nevertheless, these approaches generally treat topology as an informational feature rather than as a constrained structural object. Consequently, explicit guarantees regarding structural cohesion and communication robustness are rarely provided.

The comparative analysis highlights a persistent disconnect between structural graph analytics and intelligent swarm coordination. Existing k-truss and dynamic graph approaches provide effective mechanisms for characterizing and maintaining network cohesion, yet they remain largely external to the decision-making process. Conversely, AI-driven UAV coordination frameworks focus on adaptive behavior and mission optimization but rarely incorporate explicit mechanisms for preserving structural robustness.

As a result, current solutions are generally capable of achieving either adaptability or resilience, but seldom both simultaneously. This limitation becomes particularly pronounced in disaster-response scenarios, where communication reliability directly affects task execution, information sharing, and overall mission success.

To the best of our knowledge, no existing framework jointly integrates dynamic k-truss cohesion maintenance, agent-driven graph evolution, and decentralized AI-based coordination within a unified decision-making architecture for UAV swarms. Existing studies either treat structural cohesion as an offline analytical property or employ graph information solely as an auxiliary feature for learning.

The present work addresses this gap by embedding adaptive k-truss preservation directly into the coordination process. Rather than merely observing the communication graph, agents actively consider structural cohesion when selecting actions, thereby coupling network resilience and mission execution within a single optimization framework.

The proposed framework therefore represents a shift from passive graph-aware coordination toward structure-aware coordination, in which communication topology is not only observed but actively maintained as a first-class component of the decision-making process. This perspective provides a principled mechanism for improving robustness in dynamic and failure-prone UAV environments.

2.6. Positioning of the Proposed Approach

The proposed framework occupies a distinct position at the intersection of graph-theoretic resilience analysis and AI-driven UAV swarm coordination. Unlike traditional k-truss methods, which primarily focus on identifying or maintaining cohesive subgraphs, the proposed approach integrates structural cohesion directly into the decision-making process.

Similarly, unlike existing AI-based coordination frameworks that use graph information primarily as an input for learning or optimization, the proposed framework treats structural cohesion as an explicit operational objective. Adaptive k-truss preservation therefore becomes an active component of swarm coordination rather than a passive network property.

By coupling dynamic graph evolution, decentralized agentic decision-making, and adaptive backbone maintenance, the proposed framework enables UAV swarms to simultaneously pursue mission objectives and communication resilience. This positioning distinguishes the approach from existing graph-analytic methods and contemporary AI-based coordination strategies.

3. Theoretical Foundations and Preliminaries

This section establishes the theoretical foundations of the proposed framework by integrating concepts from graph-theoretic, dynamic network analysis, and intelligent multi-agent systems. In recent years, the convergence of graph-based modeling and artificial intelligence has led to the emergence of *graph intelligence* [25,26], where structural properties of networks are explicitly leveraged to enhance learning and decision-making processes [27]. This paradigm is particularly relevant for UAV swarm systems, where agents interact over dynamic communication topologies and must maintain coordination under uncertainty and failures.

From a systems perspective, resilience in swarm coordination is fundamentally linked to the structural properties of the underlying communication graph. Traditional connectivity measures are insufficient to capture robustness in dynamic environments, as they fail to account for redundancy and higher-order interactions. Recent studies emphasize the importance of graph-based representations and learning mechanisms for achieving scalable and adaptive coordination in swarm systems [23,28]. In this context, cohesive subgraph models provide a principled way to characterize structural robustness, motivating the adoption of k-truss-based formulations in this work.

3.1. Graph-Theoretic Modeling of Swarm Systems

We model the swarm as a time-evolving graph $G_t = (V_t, E_t)$, where nodes represent UAVs and edges represent communication links. The topology evolves due to agent mobility, environmental constraints, and link failures.

Graph-based representations have become central in modern swarm intelligence frameworks, enabling the integration of perception, communication, and control into a unified mathematical structure [29]. Moreover, recent works demonstrate that graph neural networks (GNNs) and graph-based reinforcement learning significantly improve coordination efficiency in multi-agent systems by exploiting relational inductive biases [30].

3.2. Triangle-Based Cohesion and Structural Redundancy

A key limitation of traditional graph metrics is their inability to capture higher-order interactions. Triangle-based structures provide a natural measure of local redundancy, as they encode multiple alternative communication paths.

For an edge (u, v) , the triangle support is defined as

$$\text{support}(u, v) = |\mathcal{N}(u) \cap \mathcal{N}(v)|.$$

Figure 1 illustrates this concept for an edge (u, v) with two common neighbors, w_1 and w_2 , yielding $\text{support}(u, v) = 2$. This measure reflects the robustness of a communication link: edges embedded in multiple triangles are less vulnerable to failure. Recent studies

in network resilience highlight that triangle-rich structures significantly improve fault tolerance and information diffusion stability [31].

$$\text{support}(u, v) = 2$$

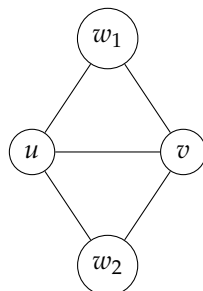


Figure 1. Triangle support as a measure of local redundancy in communication graphs.

3.3. *k*-Truss Cohesion as a Structural Resilience Model

Let $G = (V, E)$ be an undirected communication graph. For an edge $e = (u, v) \in E$, its triangle support in a subgraph $H \subseteq G$ is defined as

$$\text{supp}_H(u, v) = |N_H(u) \cap N_H(v)|, \quad (1)$$

where $N_H(u)$ denotes the neighbors of u in H . The subscript H is essential because triangle support must be evaluated with respect to the subgraph currently under consideration, rather than permanently with respect to the original graph.

Definition 1 (*k*-truss). A subgraph $H \subseteq G$ is a *k*-truss if every edge $e \in E(H)$ satisfies

$$\text{supp}_H(e) \geq k - 2. \quad (2)$$

The condition in Equation (2) provides a local redundancy requirement. Every retained communication edge belongs to at least $k - 2$ triangles inside the retained subgraph. Each such triangle contains an alternative two-hop route between the endpoints of the edge. This does not constitute a global guarantee against arbitrary simultaneous failures, but it ensures that every retained edge is embedded in a minimum number of locally redundant communication patterns. The family of trusses is nested as follows:

$$T_{k+1}(G) \subseteq T_k(G). \quad (3)$$

Consequently, increasing k strengthens the minimum triangle-support requirement while generally reducing the number of retained nodes and edges. This monotonic relation gives rise to the resilience–coverage trade-off used throughout the proposed framework: larger values of k favor stronger local redundancy, whereas smaller values preserve a broader communication backbone.

Proposition 1 (Local redundancy and backbone monotonicity). For every edge $e \in T_k(G)$, $\text{supp}_{T_k(G)}(e) \geq k - 2$. Moreover, $T_{k+1}(G) \subseteq T_k(G)$. Therefore, increasing k cannot decrease the minimum support required of a retained edge, but it cannot increase the feasible backbone edge set.

Proof. The first statement follows directly from the definition of a *k*-truss. For the second statement, every edge satisfying the $(k + 1)$ -truss condition has support at least $k - 1$, and therefore also satisfies the weaker *k*-truss requirement of support at least $k - 2$. Hence, every $(k + 1)$ -truss is contained in a *k*-truss. $\square$

Proposition 1 establishes the graph-theoretic basis of the adaptive selection mechanism introduced later. The parameter k does not directly guarantee mission success or global network stability. Instead, it controls the minimum local redundancy of the communication backbone, which in turn influences the structural features supplied to the decentralized AI decision process.

3.4. Dynamic Graphs and Structural Adaptation

In real-world swarm systems, the communication graph evolves continuously. This introduces significant challenges for maintaining structural properties such as cohesion and connectivity.

Recent works on dynamic graph intelligence emphasize the need for adaptive mechanisms that update structural representations in real time [27]. In UAV swarms, topology changes may result from mobility, environmental obstacles, or node failures, requiring continuous adaptation.

3.5. From Structural Robustness to Intelligent Coordination

The integration of structural graph properties with AI-based decision-making is an emerging research direction. Modern approaches combine graph representations with learning-based policies to achieve adaptive and scalable coordination [28,30].

However, existing methods typically treat the graph as an input feature rather than enforcing structural constraints. This limits their ability to guarantee robustness under adverse conditions.

3.6. AI-Enabled Structure-Aware Swarm Coordination

The proposed framework leverages a graph-theoretic communication infrastructure and decentralized AI decision-making to enable robust UAV swarm coordination during dynamic operations. The communication graph is used to encode structural information about the network's operation, while individual UAVs make tactical decisions based on their observations and the graph descriptors derived from it. Figure 2 provides an overview of the interaction between these components.

In this work, we use the term *Agentic AI* to refer to a specific set of engineering capabilities. In particular, each UAV in the swarm is an agent that makes decisions about its own operation, based on the information about the communications structure available to it. The agents operate without central coordination, dynamically evaluating their options and adapting their behavior. This work's key contributions are the coordinated use of these AI capabilities in concert with decentralized graph maintenance procedures. Specifically, the proposed framework has two interacting components:

- The procedure to build a dynamic communication graph and maintain its k -truss backbone, which ensures the presence of locally redundant communication paths;
- The set of graph-aware AI coordination procedures, which make it possible to incorporate the structural information into the decision making of individual agents.

The interaction between these two processes takes the form of the enhanced closed-loop coordination mechanism described in the following sections. In particular, after the agents perform coordination actions, the communication graph is updated, its backbone is rebuilt, and the descriptors updated, allowing the agents to incorporate latest information into their decision making. In this way, the graph structure becomes a critical consideration in the agents' operations, which are not optimized around static constraints, but instead dynamically adapted.

Agentic k -Truss Drone Swarm Coordination

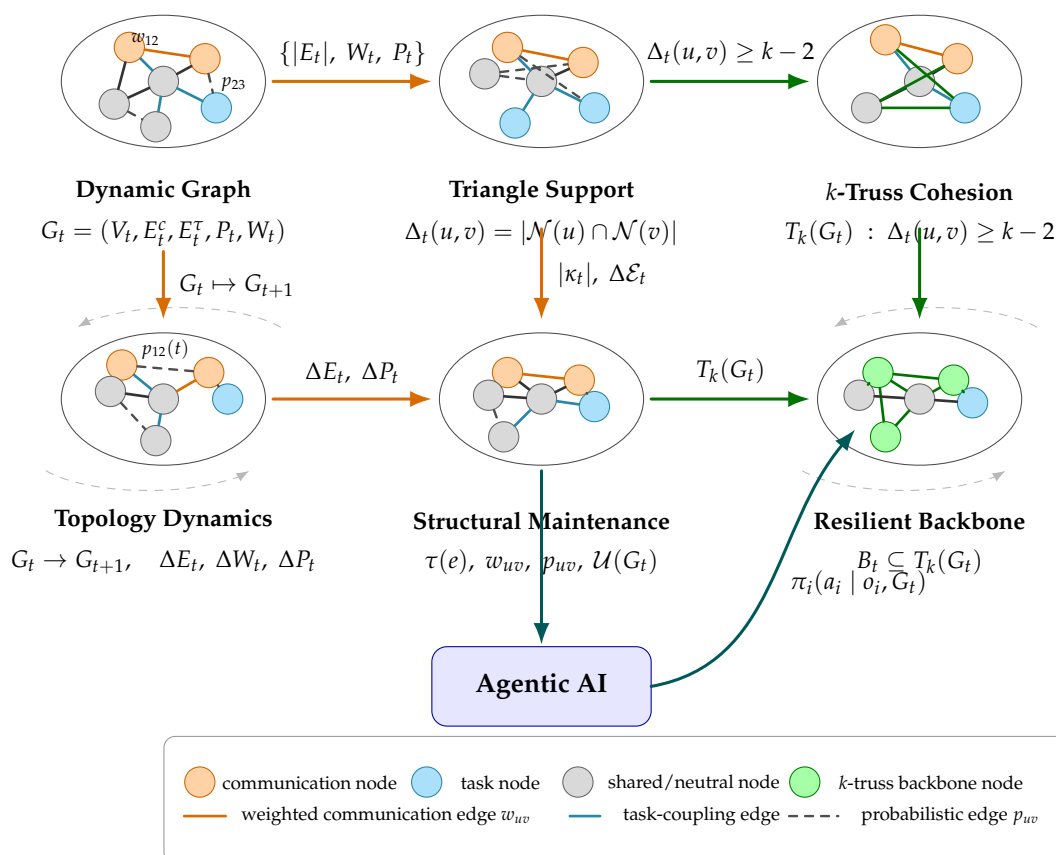


Figure 2. Multi-layer graph-based view of the proposed agentic k -truss framework. The top row extracts triangle-supported cohesion from the dynamic swarm graph, while the bottom row captures topology evolution, structure maintenance, and backbone formation. Agentic AI operates on graph state, edge weights, and uncertainty to preserve a resilient communication-task backbone.

4. Motivating Example

Motivating Example: Toward Resilient and Adaptive Drone Swarms

Natural disaster environments, such as large-scale floods, earthquakes, or wildfires, are characterized by high uncertainty, rapidly evolving conditions, and limited communication infrastructure. In such scenarios, unmanned aerial vehicle (UAV) swarms offer a promising solution due to their mobility, scalability, and ability to operate without centralized control.

We consider a disaster-response scenario in which a swarm of N UAVs is deployed to perform multiple cooperative tasks, including victim localization, environmental monitoring, and communication relaying. Each UAV operates autonomously while interacting with neighboring agents through wireless communication links. These interactions form a dynamic network whose structure directly impacts the effectiveness of coordination.

At any given time t , the swarm can be represented as a graph:

$$G_t = (V_t, E_t^c, E_t^\tau), \quad (4)$$

where V_t is the set of UAVs, E_t^c represents communication links, and E_t^τ encodes task-related interactions. This abstraction provides a unified view of both communication and collaboration within the swarm.

However, unlike idealized network models, the communication graph in disaster environments is inherently unstable. Obstacles, interference, and environmental hazards

may cause links to degrade or disappear entirely. As a result, the topology of G_t evolves continuously, often in unpredictable ways.

A critical consequence of this instability is the potential fragmentation of the swarm. For example, if a communication link (i, j) fails and no alternative paths exist between the corresponding agents, the swarm may split into disconnected components. Such fragmentation can severely degrade coordination, leading to delays in task execution or even mission failure.

This observation highlights a fundamental limitation of traditional connectivity-based models: ensuring mere connectivity is insufficient to guarantee robustness. Instead, what is needed is a notion of *redundant connectivity*, where communication links are supported by multiple alternative paths.

To capture this redundancy, we consider the concept of triangle-based connectivity. Let $\mathcal{N}_t(i)$ denote the set of neighbors of node i in the communication graph. The support of an edge (i, j) is defined as

$$s_t(i, j) = |\mathcal{N}_t(i) \cap \mathcal{N}_t(j)|, \quad (5)$$

which counts the number of common neighbors shared by i and j . Each such neighbor forms a triangle with (i, j) , providing an alternative two-hop communication path.

Edges with higher support are inherently more resilient, as the failure of a single link can be compensated by alternative routes through shared neighbors. This motivates the use of *k-truss cohesion*, which enforces that every edge is supported by at least $(k - 2)$ triangles.

While this structural constraint significantly enhances robustness, it does not fully solve the coordination problem. Indeed, maintaining a k-truss structure in a dynamic environment requires continuous adaptation. UAVs must reposition themselves, establish new links, or abandon unreliable ones in response to environmental changes.

This introduces a second layer of complexity: decision-making. Each UAV must determine its actions based on partial observations, local interactions, and the current state of the communication graph. Importantly, these actions do not merely affect individual agents—they reshape the entire network topology.

Consequently, the swarm coordination problem is inherently coupled: the graph structure constrains agent decisions, while agent decisions, in turn, modify the graph. This bidirectional dependency renders both purely structural approaches and purely learning-based approaches insufficient.

The key insight emerging from this example is that resilient swarm coordination requires a unified framework that integrates

- **Structural robustness**, ensured by k-truss-based constraints, and
- **Adaptive decision-making**, achieved through decentralized agentic policies.

As illustrated in Figure 3, the disaster-response scenario induces a dynamic communication–task graph G_t , where UAVs (agents) must simultaneously maintain reliable connectivity and execute heterogeneous tasks under uncertain and time-varying conditions. In such environments, communication links are inherently unstable due to obstacles, interference, and node mobility, leading to intermittent or failed connections, while task demands evolve dynamically as new missions (e.g., victim rescue, medical delivery, and relay support) emerge. This results in a tightly coupled interplay between the communication topology and task allocation structure, captured by the time-varying graph $G_t = (V_t, E_t^c, E_t^r, P_t, W_t)$.

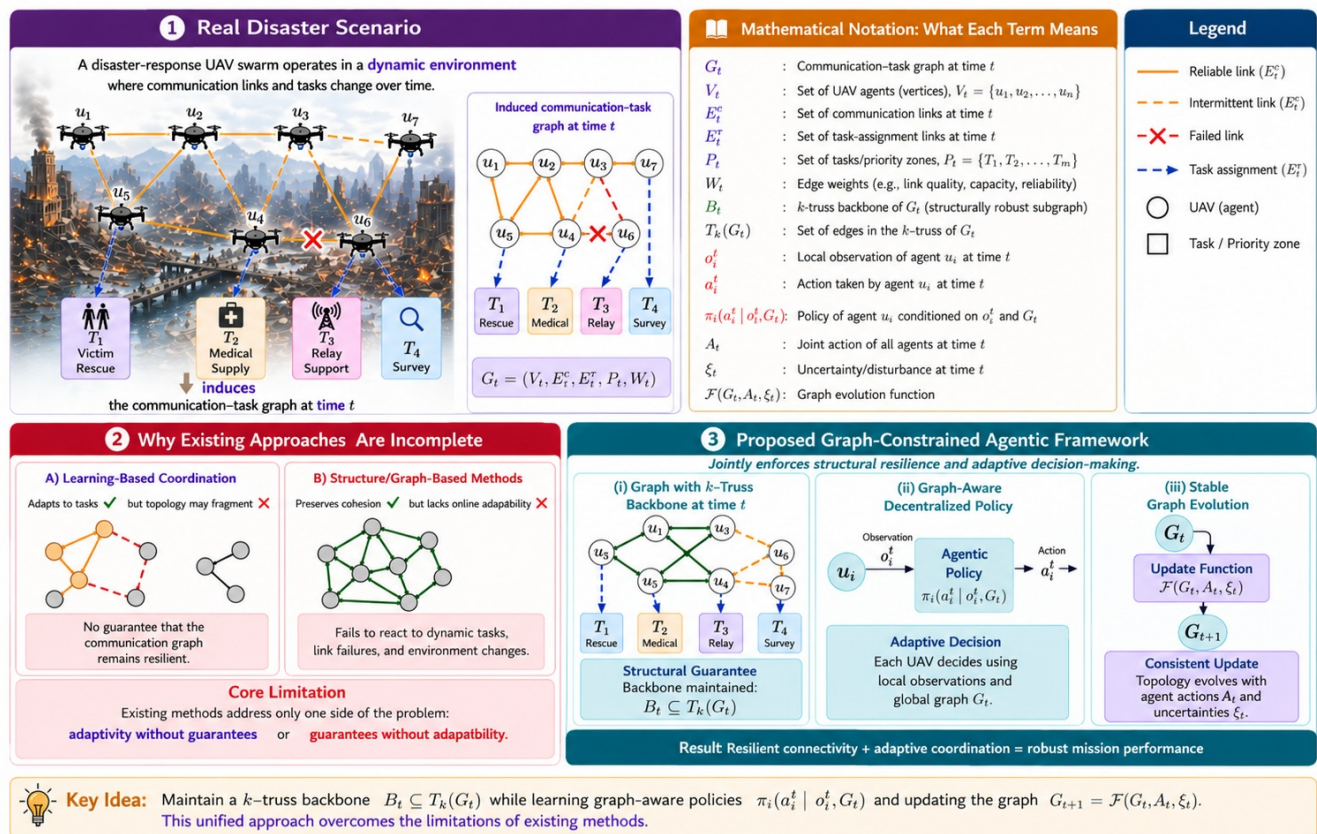


Figure 3. Motivating example: Graph-constrained agentic coordination for disaster-response UAV swarms. (1) *Real scenario*: A dynamic disaster environment induces a time-varying communication-task graph G_t . UAVs must maintain connectivity while executing heterogeneous tasks such as rescue, medical delivery, and relay support. (2) *Limitations of existing approaches*: Learning-based methods adapt to tasks but lack structural guarantees, while graph-based methods ensure cohesion but fail to adapt to dynamic conditions. (3) *Proposed framework*: The graph-constrained agentic approach jointly enforces structural resilience via a k -truss backbone $B_t \subseteq T_k(G_t)$ and adaptive decentralized decision-making through graph-aware policies $\pi_i(a_i^t | o_i^t, G_t)$, with consistent graph evolution $G_{t+1} = \mathcal{F}(G_t, A_t, \xi_t)$.

However, existing approaches remain fundamentally limited. Learning-based coordination strategies can adapt to task dynamics but often neglect structural guarantees, making the communication graph prone to fragmentation. Conversely, structure-driven methods (e.g., graph-theoretic or connectivity-preserving approaches) ensure robustness but lack the flexibility required to respond to dynamic tasks and environmental uncertainties. This dichotomy highlights a critical gap: adaptivity without guarantees versus guarantees without adaptivity.

To address this challenge, we propose a graph-constrained agentic framework that unifies both aspects. Specifically, the approach enforces structural resilience by maintaining a k -truss backbone $B_t \subseteq T_k(G_t)$ while enabling decentralized and adaptive decision-making through graph-aware policies $\pi_i(a_i^t | o_i^t, G_t)$. The system further accounts for environmental uncertainty via a consistent graph evolution process $G_{t+1} = \mathcal{F}(G_t, A_t, \xi_t)$, ensuring that topology adaptation and agent actions remain jointly optimized over time. The motivating example thus demonstrates the bidirectional interaction between communication structure and decentralized AI-enabled coordination. The communication graph restricts the set of feasible actions for the UAVs, while UAV mobility, task allocation, and relay decisions, in turn, modify the communication graph. The proposed framework decouples this coupled problem into a closed-loop coordination process in which a communication graph is

constructed, an adaptive k -truss backbone is extracted, graph-based structural descriptors are computed, decentralized graph-aware actions are executed, and the communication graph is updated. This closed-loop process is detailed in the following.

5. Problem Formulation

The motivating example in Figure 3 highlights a fundamental characteristic of disaster-response UAV swarms: coordination emerges from a continuous interplay between communication constraints and task execution in a dynamic and uncertain environment. As UAVs navigate a disrupted landscape, communication links fluctuate due to obstacles and interference, while mission objectives evolve as new tasks appear and existing ones change in priority. Consequently, the swarm cannot be adequately described by static models or decoupled formulations. Instead, it must be understood as a *time-evolving system* in which connectivity, coordination, and decision-making are intrinsically intertwined.

This observation naturally leads to a central question: *how can a distributed swarm maintain reliable coordination while adapting to continuously changing conditions?* Addressing this question requires a formulation that captures not only the structure of interactions among agents but also the dynamics that govern their evolution. In what follows, we develop such a formulation by progressively integrating communication topology, task allocation, and decentralized decision-making into a unified mathematical framework.

5.1. A Unified Graph Representation of Communication and Tasks

To faithfully capture the structure of the system, we model the swarm at time t as a dynamic communication–task graph:

$$G_t = (V_t, E_t^c, E_t^t, P_t, W_t). \quad (6)$$

This representation extends beyond conventional graph models by explicitly incorporating both communication and task-related interactions within a single structure. Each component of G_t plays a distinct and complementary role.

The set $V_t = \{u_1, \dots, u_n\}$ denotes the UAV agents, each acting as an autonomous decision-making entity. The communication edges $E_t^c \subseteq V_t \times V_t$ define which pairs of agents can exchange information at time t . These links are inherently unreliable and evolve continuously due to mobility, environmental obstacles, and signal degradation.

In parallel, the task-assignment edges $E_t^t \subseteq V_t \times P_t$ connect agents to tasks, where $P_t = \{T_1, \dots, T_m\}$ represents the set of active missions. This layer captures how responsibilities are distributed across the swarm, reflecting the operational objectives of the system. The weight set W_t further enriches the model by encoding quantitative attributes such as communication reliability, bandwidth, or cost.

Viewed holistically, G_t can be interpreted as a *multi-layer structure*: a communication layer governing information flow, and a task layer governing mission execution. Crucially, these layers are not independent. Communication constraints directly affect coordination capabilities, while task-driven movements reshape the communication topology. This mutual dependence is at the heart of the system's complexity.

Moreover, the graph evolves over time as a result of several interacting factors, including UAV mobility, environmental disturbances, and the dynamic nature of tasks. This leads to a sequence $\{G_t\}_{t=0}^T$ of time-indexed graphs, capturing the continuous transformation of the swarm.

5.2. Agent State Representation

Although coordination is performed in a decentralized manner, each UAV must maintain an internal representation of its local operational context. We therefore associate each agent u_i with a state vector

$$s_i^t = (x_i^t, e_i^t, \mathcal{N}_i^t, \mathcal{T}_i^t, \tau_i^t), \quad (7)$$

where x_i^t denotes the current position of the UAV, e_i^t represents its remaining energy, $\mathcal{N}_i^t$ denotes the set of neighboring agents within communication range, $\mathcal{T}_i^t$ represents locally observable tasks, and τ_i^t characterizes the structural support associated with the agent's communication links.

The state representation combines operational, communication, and structural information, enabling agents to reason simultaneously about mission objectives and network resilience. This information forms the basis of the observation model used during decentralized decision making.

5.3. Ensuring Structural Resilience Through k -Truss Cohesion

While connectivity is necessary for coordination, it is far from sufficient in disaster scenarios. A network that is merely connected may still be fragile, as the failure of a small number of links can disrupt global communication. What is required instead is a form of *structural resilience*, where connectivity is supported by redundancy.

To formalize this notion, we adopt the concept of k -truss, a graph-theoretic structure that enforces local cohesiveness.

Definition 2 (k -Truss). A subgraph $H \subseteq G_t$ is said to be a k -truss if every edge in H participates in at least $(k - 2)$ triangles within H .

Connectivity alone does not ensure the level of redundancy necessary for disaster scenarios. A connected graph can have bridges or other lightly supported edges whose removal disconnects the swarm. We therefore define the resilient communication backbone as:

$$B_t \subseteq T_{k_t}(G_t^c), \quad (8)$$

where $G_t^c = (V_t, E_t^c)$ is the communication layer and k_t is the truss parameter selected at time t . For every edge $e \in E(B_t)$,

$$\text{supp}_{B_t}(e) \geq k_t - 2. \quad (9)$$

Equation (9) ensures that every retained backbone edge is supported by a minimum number of common-neighbor communication patterns inside the backbone itself. The value of k_t defines a structural trade-off. Too high, and we exclude lightly supported UAVs or links; too low, and we fail to enforce a stringent minimum support condition for the backbone edges. The adaptive selector introduced in Section 6.2 therefore identifies the maximum truss level for which the backbone achieves a minimum coverage. The structural backbone is not an end in itself. The membership counts, incident support, retained edge ratios, and local connectivity properties are all fed into descriptors that influence the decentralized AI decision rule described in Section 5.4.

5.4. Decentralized Graph-Aware AI Decision Rule

Each UAV acts from locally available information. For UAV u_i , the operational observation is

$$o_i^t = (s_i^t, \{s_j^t : j \in \mathcal{N}_i^t\}), \quad (10)$$

where s_i^t contains the UAV state, locally visible tasks, residual energy, and communication-quality measurements. To expose the structural role of the UAV, we define the local graph-aware descriptor.

$$z_i^t = [d_i^t, \bar{q}_i^t, \bar{\tau}_i^t, b_i^t, \rho_i^t, \ell_i^t], \quad (11)$$

where:

- $d_i^t = |N_i^t|$ is the local communication degree;
- $\bar{q}_i^t$ is the mean quality of incident communication links;
- $\bar{\tau}_i^t$ is the mean triangle support of incident backbone edges;
- $b_i^t \in \{0, 1\}$ indicates whether u_i belongs to B_t ;
- ρ_i^t is the fraction of incident communication edges retained in B_t ;
- ℓ_i^t is the normalized size of the connected component containing u_i .

The complete graph-aware input is

$$h_i^t = \Phi(o_i^t, z_i^t), \quad (12)$$

where Φ denotes feature concatenation and normalization in the current implementation. No centralized global graph embedding is required; each UAV uses only locally available neighborhood and backbone information. The decentralized AI decision rule is deterministic:

$$\mu_i(o_i^t, z_i^t) = \arg \max_{a \in \mathcal{A}_i^t} U_i^t(a), \quad (13)$$

where $\mathcal{A}_i^t$ is the feasible action set and $U_i^t(a)$ is the local multi-objective utility defined in Equation (27). Thus, the AI contribution of the current framework lies in graph-aware autonomous action evaluation and selection, rather than in a newly proposed reinforcement-learning training procedure. The joint action is

$$A_t = (a_1^t, \dots, a_n^t). \quad (14)$$

5.5. Closed-Loop Interaction Between Actions and Structure

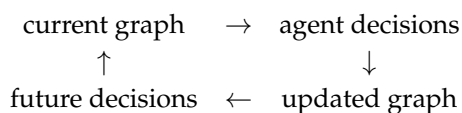
A defining feature of the system is the bidirectional relationship between agent actions and graph structure. Actions taken by agents—such as moving closer to a task or repositioning to maintain connectivity—directly influence the communication topology. In turn, the resulting topology shapes the decisions that agents can make in the future.

We capture this interaction through the graph evolution equation:

$$G_{t+1} = \mathcal{F}(G_t, A_t, \xi_t), \quad (15)$$

where $\mathcal{F}$ represents the transition dynamics and ξ_t models stochastic disturbances.

This formulation reveals the inherently sequential nature of the problem. The swarm operates in a closed loop:



As a result, coordination must be understood not as a static optimization problem but as a dynamic process unfolding over time.

5.6. Resilience-Aware Multi-Objective Optimization

The ultimate objective of the proposed framework is to enable UAV swarms to accomplish mission objectives efficiently while preserving communication resilience and

structural robustness in the presence of dynamic environmental changes, communication failures, and evolving task demands.

Achieving this objective requires balancing several potentially conflicting requirements. Agents should prioritize mission execution, maintain communication connectivity, preserve structurally cohesive network configurations, and avoid excessive resource consumption. Consequently, the coordination problem is naturally formulated as a multi-objective optimization problem.

To capture these requirements, the instantaneous reward at time t is defined as

$$R_t = \lambda_1 S_t + \lambda_2 C_t + \lambda_3 T_t - \lambda_4 E_t, \quad (16)$$

where

- S_t denotes the task-performance component,
- C_t denotes the communication-connectivity component,
- T_t denotes the structural-cohesion component,
- E_t denotes the energy-consumption penalty.

The coefficients $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$ control the relative importance of mission effectiveness, communication resilience, structural robustness, and operational efficiency. The coefficients $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are mission-level design parameters that are fixed during each simulation episode. By keeping the reward weights fixed during an episode, the effect of graph evolution, adaptive backbone reconstruction, and decentralized graph-aware decision-making are isolated. Since the four reward components are normalized, the weights satisfy

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1, \quad \lambda_r \geq 0, \quad (17)$$

thereby ensuring a consistent interpretation of the relative importance of mission effectiveness, communication robustness, structural integrity and energy efficiency. The impact of different weights are analyzed in the reward-weight sensitivity analysis in Section 7.9. The task-performance component is defined as

$$S_t = \frac{\sum_{p \in P_t} w_p c_p}{\sum_{p \in P_t} w_p}, \quad (18)$$

where P_t denotes the set of active tasks, w_p represents the priority weight associated with task p , and $c_p \in \{0, 1\}$ indicates whether the task has been successfully completed.

This term rewards successful completion of mission objectives while giving greater importance to high-priority disaster-response tasks.

The communication-connectivity component is defined as

$$C_t = \frac{|LCC_t|}{|V_t|}, \quad (19)$$

where $|LCC_t|$ denotes the size of the largest connected component of the communication graph at time t and $|V_t|$ is the total number of active UAVs.

This term encourages preservation of global communication connectivity and penalizes network fragmentation.

The structural-cohesion component is derived from the adaptive k -truss backbone and is defined as

$$T_t = \frac{|V(B_t)|}{|V_t|}, \quad (20)$$

where B_t denotes the current resilient backbone maintained within the communication graph.

This term explicitly rewards the preservation of structurally cohesive communication patterns and promotes triangle-supported redundancy throughout the network.

Finally, the resource-consumption penalty is defined as

$$E_t = \frac{1}{|V_t|} \sum_{i=1}^{|V_t|} e_i^{\text{used}}, \quad (21)$$

where e_i^{used} represents the energy consumed by UAV u_i during the current time step. This penalty discourages excessive mobility and communication overhead, thereby improving operational efficiency and mission endurance.

The multi-objective criterion used to evaluate candidate actions at each coordination step is defined by the preceding components. In the ATAC controller, each drone uses its current local observations and structural information from the graph to perform a constrained utility maximization in a single step. For drone u_i at coordination step t , let $\mathcal{A}_i^t$ be the set of feasible actions given the current constraints of mobility, communication, task, safety, and structure. The local utility associated with the candidate action $a \in \mathcal{A}_i^t$ is denoted as follows:

$$U_i^t(a) = \lambda_1 S_i^t(a) + \lambda_2 C_i^t(a) + \lambda_3 T_i^t(a) - \lambda_4 E_i^t(a), \quad (22)$$

where the four terms represent the action-dependent counterparts of the task-performance, communication-connectivity, structural-cohesion, and resource-consumption components defined above. The implemented decentralized decision rule is therefore

$$a_i^{t,*} = \arg \max_{a \in \mathcal{A}_i^t} U_i^t(a). \quad (23)$$

Equation (23) defines a one-step constrained optimization rather than an explicit optimization of an expected cumulative discounted return. Once the selected actions are executed, the UAV states and communication topology are updated, the adaptive k -truss backbone and graph-derived descriptors are recomputed, and the decision rule is applied again at the next coordination step. The controller is therefore myopic with respect to its optimization horizon, but closed-loop and adaptive with respect to the evolving mission and communication state. This formulation is adopted to preserve decentralized online tractability. Explicit multi-step optimization over future joint swarm trajectories would require a predictive transition model and reasoning over the coupled future actions of multiple UAVs, as well as future topology changes, failures, and task evolution. These mechanisms are not assumed in the present implementation. Accordingly, the implemented decision rule is not claimed to solve, or to provide a formal approximation guarantee for, a separate cumulative discounted-return optimization problem. The coefficients $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are thus mission-design parameters of the online multi-objective utility rather than learned policy parameters. Their influence on the resulting mission–structure–resource operating point is evaluated empirically in the reward-weight sensitivity analysis in Section 7.9.

5.7. Formal Problem Statement

The preceding formulation establishes a unified view of UAV swarm coordination as a dynamic graph optimization problem in which communication structure and agent behavior evolve jointly over time. The objective is not only to maximize mission effectiveness but also to preserve the structural resilience required for sustained coordination under adverse operating conditions.

Accordingly, the problem addressed in this work can be stated as follows:

Problem Statement. Given the communication–task graph G_t at coordination step t , its adaptive k -truss backbone B_t , the locally available state of each UAV, and the corresponding feasible action set $\mathcal{A}_i^t$, determine for each active UAV u_i the decentralized action

$$a_i^{t,*} = \arg \max_{a \in \mathcal{A}_i^t} U_i^t(a), \quad (24)$$

while

1. Maintaining an adaptive k -truss-supported communication backbone under the current topology;
2. Supporting efficient execution of dynamically evolving mission tasks;
3. Preserving useful communication connectivity under stochastic link and UAV failures;
4. Balancing mission effectiveness, structural cohesion, and resource consumption.

The selected actions modify UAV positions, task states, and potentially the communication topology. The graph is therefore updated after each coordination step, after which the backbone and graph-derived descriptors are recomputed and the decision process is repeated. The resulting formulation is a repeated closed-loop decision process rather than a single optimization over the complete mission horizon.

This problem statement provides the foundation for the adaptive graph-constrained coordination framework presented in the following section.

6. Proposed Approach: Graph-Constrained Agentic Coordination

The formulation developed in Section 5 establishes UAV swarm coordination as a closed-loop dynamic graph decision problem in which communication structure and decentralized actions evolve jointly over time. In disaster-response environments, communication links are continuously affected by mobility, environmental obstacles, interference, and failures, while mission requirements change as new tasks emerge and operational priorities evolve. Consequently, effective coordination requires both responsive decentralized decision-making and preservation of the structural properties that support reliable information exchange across the swarm. To address this challenge, we propose a resilience-aware graph-constrained agentic coordination framework that integrates structural cohesion preservation directly into the decision-making process. Unlike conventional approaches that treat communication topology as a passive support mechanism, the proposed framework explicitly maintains an adaptive k -truss backbone that serves as a resilient communication core while simultaneously guiding decentralized agent actions. In this way, mission execution and communication resilience become tightly coupled objectives rather than independent optimization goals.

The framework consists of four interacting components: (i) dynamic communication–task graph construction, (ii) adaptive k -truss backbone extraction, (iii) graph-aware agentic decision-making, and (iv) closed-loop graph evolution. Together, these components establish a continuous feedback process in which graph structure influences agent behavior, while agent actions actively reshape and preserve the underlying communication topology.

Figure 4 illustrates the proposed framework in a representative disaster-response scenario. The following subsections describe each component in detail and present the algorithms that implement the proposed coordination strategy.

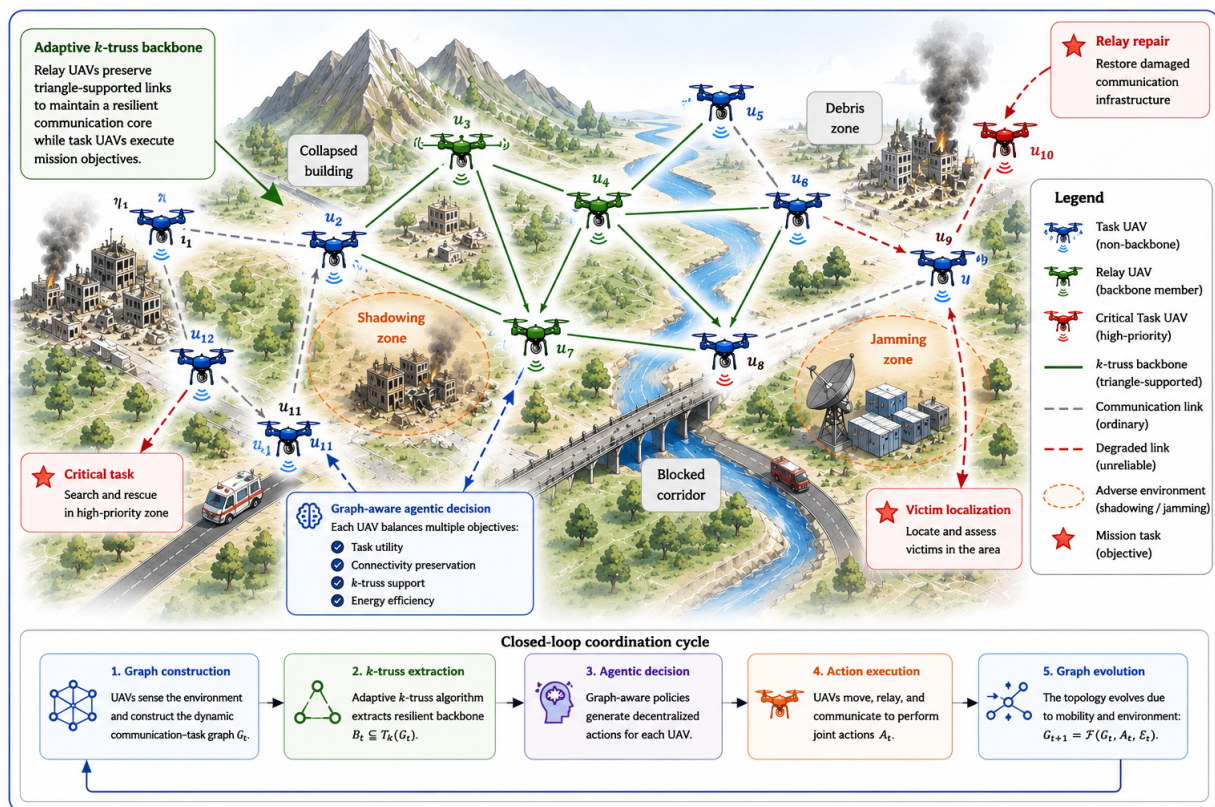


Figure 4. Overview of the proposed resilience-aware graph-constrained agentic coordination framework in a representative disaster-response scenario. UAVs perform mission tasks while maintaining an adaptive k -truss communication backbone that preserves structural resilience under dynamic topology changes, communication disruptions, and environmental constraints.

The scenario illustrated in Figure 4 captures the central principle of the proposed approach: communication structure and agent behavior evolve jointly and must therefore be managed within a unified coordination framework. UAVs operate in a dynamic disaster environment characterized by evolving tasks, communication disruptions, environmental obstacles, and potential node failures. To maintain mission continuity under such conditions, the swarm continuously constructs a communication–task graph, extracts a resilient adaptive k -truss backbone, and generates graph-aware decisions that balance task execution with communication resilience.

This closed-loop interaction enables the swarm to preserve triangle-supported communication structures while adapting to changing operational requirements. The following subsections describe the four core components of the framework and the algorithms that realize this coordination process.

6.1. Dynamic Communication–Task Graph Construction

The first component transforms the physical disaster-response environment into the graph representation required by the proposed framework. This step is essential because UAV positions, communication conditions, and task locations evolve over time; therefore, the graph cannot be assumed static or fixed.

At each time step t , the swarm is represented by $G_t = (V_t, E_t^c, E_t^r, P_t, W_t)$. The communication layer E_t^c captures feasible UAV-to-UAV links, while the task layer E_t^r connects UAVs to active tasks in P_t . The weights W_t encode link quality, reliability, cost, or other communication attributes relevant to coordination.

Algorithm 1 summarizes this construction process.

Algorithm 1 Dynamic Communication–Task Graph Construction**Require:** UAV states, task set P_t , communication thresholds**Ensure:** Dynamic graph G_t

```

1: Initialize  $V_t, E_t^c \leftarrow \emptyset, E_t^r \leftarrow \emptyset$ 
2: for each pair of UAVs  $(u_i, u_j)$  do
3:   if communication feasibility condition is satisfied then
4:     Add  $(u_i, u_j)$  to  $E_t^c$ 
5:     Assign edge weight  $w_{ij}^t$ 
6:   end if
7: end for
8: for each UAV  $u_i$  and task  $T_k \in P_t$  do
9:   if task-assignment condition is satisfied then
10:    Add  $(u_i, T_k)$  to  $E_t^r$ 
11:   end if
12: end for
13: return  $G_t = (V_t, E_t^c, E_t^r, P_t, W_t)$ 

```

This graph becomes the input to the structural-resilience layer. However, the constructed graph may still contain fragile links; therefore, a second step is required to extract a robust communication backbone.

6.2. Structural Backbone Extraction via k -Truss Cohesion

Once G_t is constructed, the next objective is to identify the part of the communication graph that is structurally reliable. Simple connectivity is not enough in disaster environments because a connected graph may still fragment after a small number of link failures.

To overcome this weakness, we extract a k -truss backbone $B_t \subseteq T_k(G_t)$. This backbone retains edges that have sufficient triangle support, meaning that each retained link is reinforced by alternative local paths. Figure 5 illustrates this pruning process before Algorithm 2 formalizes it.

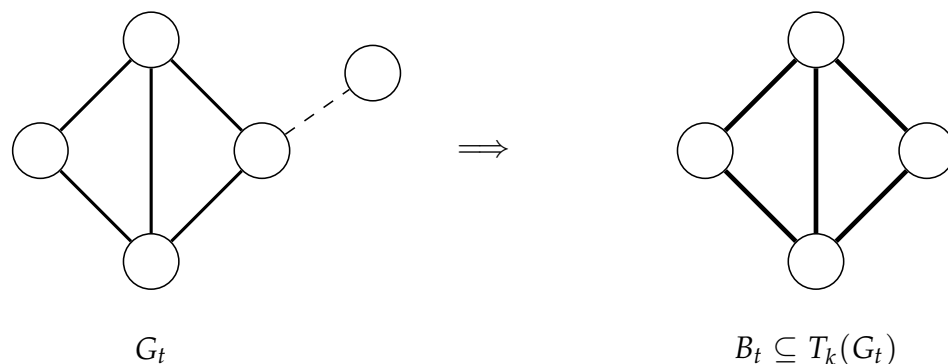


Figure 5. Illustration of k -truss backbone extraction. Weak edges with insufficient triangle support are removed, leaving a structurally cohesive backbone B_t .

Algorithm 2 evaluates and updates support with respect to the current residual graph B . When an edge is removed, every triangle containing that edge is destroyed, and the support of the two remaining triangle edges is decremented. Consequently, when the queue becomes empty,

$$\text{supp}_{B_t}(e) \geq k_t - 2, \quad \forall e \in E(B_t), \quad (25)$$

and the returned backbone satisfies the standard k -truss definition. Algorithm 2 is a support-based extraction procedure, not a newly proposed incremental truss-maintenance algorithm. Dynamic adaptation occurs at the framework level: mobility, link formation,

link loss, and UAV failures change G_t^c , after which the backbone is recomputed from the updated graph. Existing incremental or fully dynamic truss-maintenance procedures could replace this extraction step in future large-scale implementations without changing the surrounding AI coordination architecture.

Algorithm 2 k -Truss Backbone Extraction on the Current Graph

Require: Communication graph $G_t^c = (V_t, E_t^c)$; truss parameter k_t

Ensure: Valid backbone $B_t \subseteq T_{k_t}(G_t^c)$

```

1:  $B \leftarrow G_t^c$ 
2: for all  $e = (u, v) \in E(B)$  do
3:    $\text{supp}_B(e) \leftarrow |N_B(u) \cap N_B(v)|$ 
4: end for
5:  $\mathcal{Q} \leftarrow \{e \in E(B) : \text{supp}_B(e) < k_t - 2\}$ 
6: while  $\mathcal{Q} \neq \emptyset$  do
7:   Remove an edge  $e = (u, v)$  from  $\mathcal{Q}$ 
8:   if  $e \notin E(B)$  then
9:     continue
10:  end if
11:   $\mathcal{W} \leftarrow N_B(u) \cap N_B(v)$ 
12:  Remove  $e$  from  $B$ 
13:  for all  $w \in \mathcal{W}$  do
14:    for all  $e' \in \{(u, w), (v, w)\} \cap E(B)$  do
15:       $\text{supp}_B(e') \leftarrow \text{supp}_B(e') - 1$ 
16:      if  $\text{supp}_B(e') < k_t - 2$  then
17:        Insert  $e'$  into  $\mathcal{Q}$ 
18:      end if
19:    end for
20:  end for
21: end while
22: Remove isolated vertices from  $B$ 
23:  $B_t \leftarrow B$ 
24:
25: return  $B_t$ 

```

Adaptive Selection of the Truss Parameter

Let the set of candidates be $\mathcal{K} = \{k_{\min}, \dots, k_{\max}\}$. At each coordination cycle, ATAC selects the strongest feasible truss level whose node coverage remains above a prescribed threshold $\eta \in (0, 1]$:

$$\text{added}k_t^* = \max \left\{ k \in \mathcal{K} : \frac{|V(T_k(G_t^c))|}{|V_t|} \geq \eta \right\}. \quad (26)$$

If no candidate satisfies the coverage condition, the selector uses $k_{\min}$. Equation (26) operationalizes the trade-off is stated in Proposition 1. Under A topology that is well supported would then allow a larger value of k_t^* to be chosen. After severe edge or node loss, the selector may temporarily decrease. To keep adequate backbone coverage, we should use k_t^* .

6.3. Graph-Aware Agentic Decision-Making

After the resilient backbone has been identified, each UAV must choose actions that serve the mission while preserving the communication structure. This is where the agentic component enters: agents do not merely react to local observations but reason over the graph context.

For each UAV u_i , the decision policy is conditioned on its local observation o_i^t and the graph state. In practice, the agent uses information from both G_t and B_t to infer whether its movement or task selection may strengthen or weaken the swarm structure.

Figure 6 summarizes this decision module.

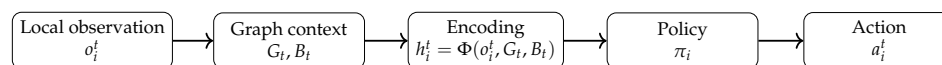


Figure 6. Graph-aware agentic decision module. Each UAV combines local observations with graph and backbone information to generate an action that supports both task execution and structural resilience.

Algorithm 3 implements the per-step constrained utility maximization in Equation (23). It is therefore the online decision rule of the closed-loop ATAC controller, rather than a solver or approximation algorithm for a separate cumulative discounted-return objective.

Algorithm 3 Graph-Aware Decentralized AI Action Selection

Require: UAV u_i ; local observation o_i^t ; graph G_t^c ; backbone B_t ; feasible action set $\mathcal{A}_i^t$

Ensure: Selected action a_i^t

- 1: Compute local degree d_i^t
 - 2: Compute mean incident-link quality $\bar{q}_i^t$
 - 3: Compute mean incident-edge support $\bar{\tau}_i^t$
 - 4: Determine backbone membership b_i^t
 - 5: Compute retained-edge ratio ρ_i^t
 - 6: Compute normalized local-component size ℓ_i^t
 - 7: $z_i^t \leftarrow [d_i^t, \bar{q}_i^t, \bar{\tau}_i^t, b_i^t, \rho_i^t, \ell_i^t]$
 - 8: $h_i^t \leftarrow \Phi(o_i^t, z_i^t)$
 - 9: **for all** $a \in \mathcal{A}_i^t$ **do**
 - 10: Predict the local effect of a on task progress, connectivity, structural support, and energy
 - 11: Evaluate $U_i^t(a)$ using Equation (27)
 - 12: **end for**
 - 13: $a_i^t \leftarrow \arg \max_{a \in \mathcal{A}_i^t} U_i^t(a)$
 - 14:
 - 15: **return** a_i^t
-

Algorithm 3 is the artificial intelligence (AI) decision-making component of ATAC. It is autonomous and “graph aware”, though in a completely deterministic way in the current implementation—the algorithm does not involve policy-gradient updates, replay-buffer sampling, actor-critic networks, or other common machine learning tools. In other words, the name AI-enabled is used here in the sense of decentralized decision making via graph-aware local rules. The graph awareness is built in by design, via the explicitly defined bonus z_i^t in the action preference mask. For instance, a higher structural score can be expected for a relay-positioning action, if it creates new incident triangle supports or reconnects a UAV to the backbone. Symmetrically, a task-directed movement might be penalized as it could disconnect a critical backbone edge, thus isolating one or more UAVs. In this way, the local processing of each UAV is linked to the global properties of a k -truss.

6.4. Resilience-Aware Decision Utility

To ensure that decentralized decisions remain aligned with the global coordination objective, each agent evaluates candidate actions through a resilience-aware utility function.

For an agent u_i , the utility of an action a_i^t is defined as

$$U_i(a_i^t) = \alpha S_i^t + \beta C_i^t + \gamma T_i^t - \delta E_i^t, \quad (27)$$

where

- S_i^t quantifies task-related benefit;
- C_i^t quantifies local connectivity preservation;
- T_i^t quantifies contribution to k-truss backbone stability;
- E_i^t quantifies energy expenditure.

The coefficients $\alpha, \beta, \gamma, \delta$ control the relative importance of mission accomplishment, communication resilience, structural cohesion, and resource efficiency. The policy therefore favors actions that improve mission performance while preserving the integrity of the resilient communication backbone. This mechanism creates a direct link between local decision-making and global structural robustness.

6.5. Closed-Loop Graph-Constrained Coordination

The previous components now come together into a closed-loop coordination process. At each time step, the graph is reconstructed, the resilient backbone is updated, agents select graph-aware actions, and the environment evolves.

This final step is essential because the system is dynamic: actions affect future connectivity, task allocation, and resilience. The method therefore does not solve a one-time graph problem; it continuously maintains a resilient and adaptive swarm.

Algorithm 4 summarizes the full proposed framework.

Algorithm 4 Closed-Loop ATAC Coordination Framework

Require: Initial environment state; candidate set $\mathcal{K}$; coverage threshold η ; horizon T

Ensure: Joint-action sequence $\{A_t\}_{t=0}^T$

- 1: Initialize G_0
- 2: **for** $t = 0$ to T **do**
- 3: Observe UAV states, active tasks, link quality, and failures
- 4: Identify topology changes

$$\Delta G_t = (\Delta E_t^+, \Delta E_t^-, \Delta V_t^-)$$

- 5: Construct the current graph G_t using Algorithm 1
 - 6: Select k_t^* using Equation (26)
 - 7: Extract B_t using Algorithm 2
 - 8: **for all** $u_i \in V_t$ **do**
 - 9: Compute a_i^t using Algorithm 3
 - 10: **end for**
 - 11: Form $A_t = (a_1^t, \dots, a_n^t)$
 - 12: Execute A_t
 - 13: Update $G_{t+1} = \mathcal{F}(G_t, A_t, \xi_t)$
 - 14: **end for**
 - 15:
 - 16: **return** $\{A_t\}_{t=0}^T$
-

6.6. Computational Complexity Analysis

The computational cost of the proposed framework arises primarily from three operations: graph construction, k-truss backbone maintenance, and decentralized decision generation.

For a swarm containing n UAVs and m communication edges, graph construction requires evaluating pairwise communication feasibility and therefore incurs a worst-case complexity of

$$O(n^2). \quad (28)$$

The k -truss extraction procedure iteratively evaluates triangle support for communication edges. Using standard support-based maintenance methods, the worst-case complexity is

$$O(m^{1.5}), \quad (29)$$

although practical performance is significantly lower for sparse UAV communication networks.

For decentralized decision-making, each UAV performs local graph-aware reasoning using information obtained from neighboring agents. Assuming bounded neighborhood size, the resulting complexity scales approximately linearly with the swarm size:

$$O(n). \quad (30)$$

Consequently, the overall complexity of each coordination cycle is dominated by graph maintenance operations, yielding

$$O(n^2 + m^{1.5}). \quad (31)$$

This complexity remains practical for medium- and large-scale UAV swarms and is further mitigated by the decentralized nature of the decision process.

6.7. Communication Overhead Analysis

Communication efficiency is an important consideration for UAV swarm coordination because excessive message exchange may increase latency, energy consumption, and network congestion.

In the proposed framework, each UAV exchanges information only with agents located within its communication neighborhood. Let d_i^t denote the degree of UAV u_i at time t . The communication cost associated with local information sharing is therefore

$$O\left(\sum_{i=1}^n d_i^t\right) = O(m). \quad (32)$$

Because coordination decisions rely primarily on local graph information and backbone membership indicators, the framework avoids global network broadcasts and centralized aggregation procedures.

As a result, communication overhead grows proportionally to the number of active communication links rather than the size of the global state space, making the approach suitable for dynamic and bandwidth-limited disaster-response environments.

Overall, the proposed framework operationalizes the central premise of this work: resilient UAV swarm coordination requires the joint consideration of structural robustness and adaptive decision-making. The dynamic communication–task graph provides a unified representation of the evolving operational environment, while the adaptive k -truss backbone preserves structurally cohesive communication patterns capable of tolerating disruptions and link failures.

At the decision level, graph-aware agentic policies exploit both local observations and structural context to generate actions that balance mission effectiveness, connectivity preservation, and resource efficiency. The resulting closed-loop interaction between graph evolution and decentralized decision-making enables the swarm to adapt continuously to changing conditions while maintaining communication resilience.

The next section evaluates the proposed framework through a comprehensive simulation campaign designed to assess mission performance, structural robustness, scalability, and fault tolerance under realistic disaster-response conditions.

7. Simulation-Based Evaluation

This section provides an evaluation of the proposed Adaptive k -Truss-Constrained Agentic AI Coordination framework (ATAC) through a series of simulation studies. Each simulation study is designed to evaluate a particular claim made in the problem formulation or methodology. All results are obtained using a custom simulation environment. As such, this section uses the terms *simulation study*, *simulation scenario*, and *simulation run* interchangeably. The current evaluation does not involve real-flight tests or hardware-in-the-loop simulations. The simulation campaign is organized around the following research questions:

- **RQ1:** Does adaptive k -truss backbone reconstruction reduce communication fragmentation and preserve local structural redundancy under stochastic communication-link failures?
- **RQ2:** Does the closed-loop graph evolution mechanism reduce structural recovery time after UAV failures?
- **RQ3:** How does the truss parameter k affect the resilience–coverage trade-off, and does adaptive selection provide a more balanced operating point than fixed- k alternatives?
- **RQ4:** Which components of the proposed framework are responsible for the observed structural and mission-level effects?
- **RQ5:** How does the computational cost evolve with the swarm size within the evaluated simulation range?
- **RQ6:** How sensitive is the framework to the mission-level reward weights under different disaster intensities?

7.1. Simulation Environment and Setup

All simulation runs were carried out in the custom simulation framework implemented for this work, which encodes the communication–task graph formulation of Section 5 along with Algorithms 1–4. It captures the time evolution of the UAV communication graph, adaptive k -truss backbone extraction, graph-aware agentic decision making, and closed-loop graph updates in the presence of stochastic mission disturbances. The simulated environment consists of a two-dimensional disaster area of size $200 \times 200 \text{ m}^2$. UAVs are initialized uniformly at random over the environment in order to simulate distributed search and rescue operations; a more centralized formation would be at odds with realistic settings. Communication links are formed between any two UAVs whose distance is below the communication radius chosen for the simulator. Tasks are placed randomly in the environment and considered complete when a UAV spends the required dwell time inside the task’s region. The simulation faithfully tracks UAV movements, communication-link formation and breakage, task allocation and completion, graph updates, and adaptive backbone maintenance in response to link failures, random UAV losses, and energy consumption. The communication graph is reworked after each coordination step; the adaptive k -truss backbone is then extracted and graph-aware decisions are made for the next step. Unless otherwise noted, all experiments were run using multiple random seeds and report mean values across independent runs. The same simulator code was used to generate all results in this section. The experimental campaign comprises six studies: (i) communication-link failures, (ii) UAV node failures, (iii) adaptive k selection, (iv) ablation, (v) computational scalability, and (vi) reward-weight sensitivity. They jointly evaluate the efficacy of the

proposed approach in terms of mission performance, communication robustness, and computational complexity. Table 3 summarises the main simulation parameters.

Table 3. Main simulation parameters.

Parameter	Symbol	Value
Arena size	L	$200 \times 200 \text{ m}^2$
Communication range	r_c	50 m
UAV maximum speed	$v_{\max}$	3.0 m/step
Task service radius	r_t	6 m
Initial number of tasks	$ \mathcal{P}_0 $	20
Maximum active tasks	$ \mathcal{P} _{\max}$	60
Task spawning probability	p_{spawn}	0.085
Simulation horizon	T	100 steps
Default swarm size	n	50 UAVs
Scalability range	—	20–90 UAVs
Initial energy	E_0	100 units
Movement energy cost	—	0.045 units/step
Communication energy cost	—	0.007 units/message
Default k	k	3
Adaptive k	—	Enabled ($k \in \{3, 4, 5\}$)
Minimum backbone coverage	η	50%
Failure injection step	t_f	25
Default link failure probability	p_{fail}	0.15
Default node failure probability	p_{node}	0.00 (experiment dependent)
Random seed	—	Experiment dependent

Figure 7 illustrates representative snapshots of the simulated environment: the healthy swarm at $t = 0$, the communication graph immediately after a node failure event, and the same swarm after ATAC has had time to restore triangle-supported backbone edges. Figure 8 illustrates the k -truss backbone extraction procedure described in Algorithm 2.

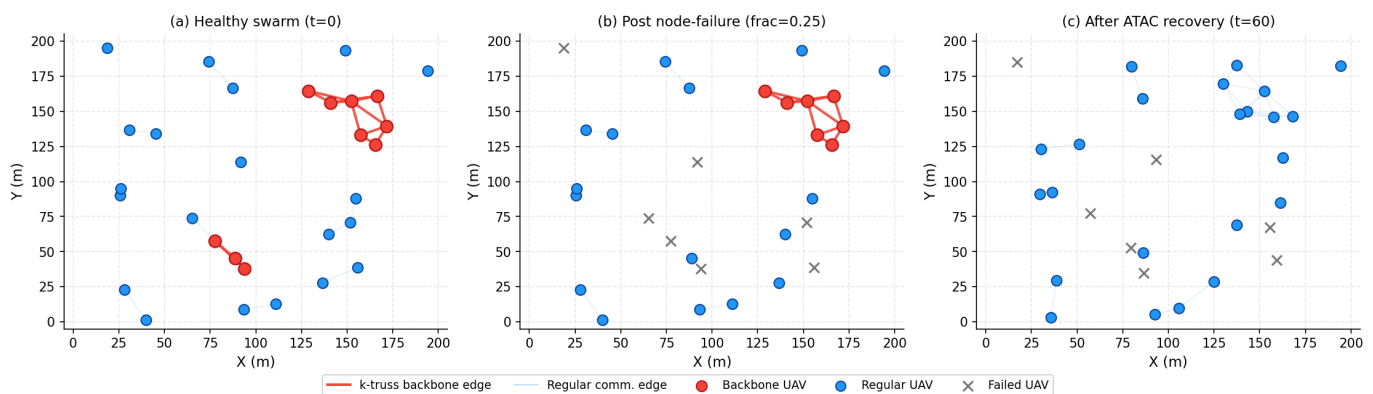


Figure 7. Simulation environment snapshots. (a) Healthy swarm at $t = 0$: all UAVs active, backbone edges (red) and regular communication edges (light blue) visible. (b) Immediately after a node-failure event ($f = 0.25$): failed UAVs marked with crosses, triangle support reduced. (c) After ATAC recovery ($t = 60$): backbone edges partially restored by repositioning surviving UAVs. Blue nodes: regular UAVs; red nodes: backbone UAVs; grey crosses: failed UAVs.

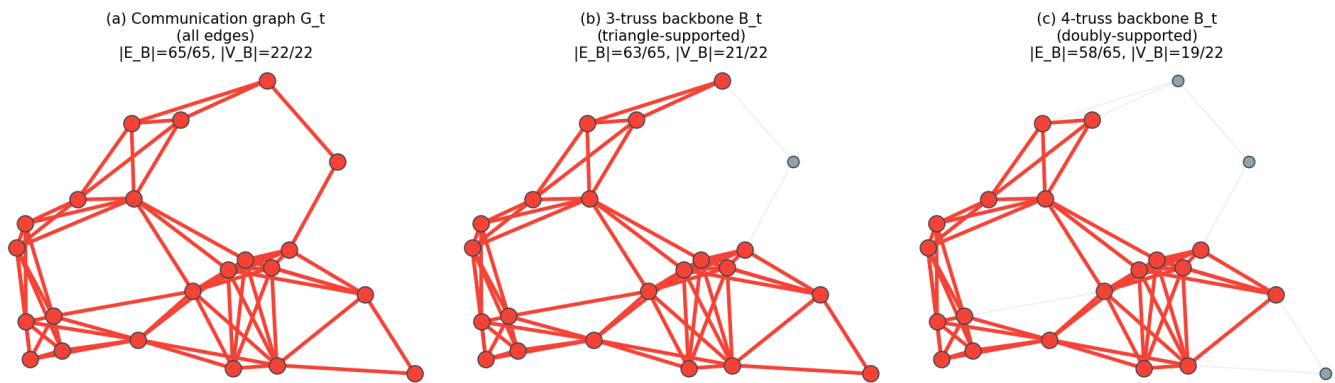


Figure 8. k -truss backbone extraction (Algorithm 2) applied to a 22-node communication graph. (a) Full communication graph G_t ; (b) 3-truss backbone: edges with triangle support ≥ 1 retained; (c) 4-truss backbone: stricter triangle requirement further prunes fragile edges. Red edges and nodes belong to the backbone; grey elements are excluded.

7.2. Compared Methods

To benchmark our approach, we evaluated several alternative coordination strategies. We mapped each baseline directly to the core components of our adaptive framework, establishing a structured and interpretable basis for the ablation study presented in the Section 7.7.

- **GTA (Greedy Task Allocation):** Actions maximise the task-performance term S_i^t in Equation (27) alone; no connectivity or structural terms. Equivalent to setting $\lambda_2 = \lambda_3 = 0$ in Equation (16).
- **CPC (Connectivity-Preserving Coordination):** Augments GTA with a connectivity penalty ($\lambda_2 > 0, \lambda_3 = 0$). Preserves the C_t term in Equation (19) but does not enforce triangle-supported cohesion.
- **GAC (Graph-Aware Coordination):** Conditions agent policies on the local graph state G_t but does not extract or maintain the k -truss backbone B_t . Structurally equivalent to ATAC with Algorithm 2 removed.
- **ATAC-Fixed:** The complete framework with the truss parameter fixed at $k = 3$ throughout each episode. Uses Algorithms 1–4 but does not adapt k .
- **ATAC (proposed):** The full framework combining dynamic graph construction (Algorithm 1), adaptive k -truss backbone extraction (Algorithm 2), graph-aware agentic decision-making (Algorithm 3), and closed-loop graph evolution (Algorithm 4), with the multi-objective reward R_t (Equation (16)).

7.3. Evaluation Metrics

To provide a comprehensive assessment of the proposed ATAC framework, the evaluation metrics are selected so that each one measures a specific component of the optimization objective defined in Equation (16). Rather than relying on a single performance indicator, the metrics jointly evaluate mission effectiveness, structural resilience, and computational efficiency.

7.4. RQ1: Resilience Under Communication-Link Failures

RQ1 examines how the evaluated coordination strategies respond when inter-UAV communication becomes progressively less reliable. This condition is particularly relevant to disaster-response operations, where mobility, obstruction, interference, and infrastructure damage can make communication intermittent even when the UAVs themselves remain operational. The purpose of this simulation study is not to reproduce a particular

disaster or wireless propagation environment but to provide a controlled assessment of how the coordination strategies respond to increasing communication degradation.

We consider five communication-link failure probabilities,

$$p_{\text{fail}} \in \{0.05, 0.12, 0.18, 0.24, 0.32\},$$

and compare GTA, CPC, GAC, ATAC-Fixed, and the proposed adaptive ATAC. Each method–failure combination is evaluated using 10 independent random seeds and 270 coordination steps. To characterize established swarm behavior rather than the initial transient, the reported statistics are computed over the final 25% of each simulation. For each seed, a metric is first averaged over this interval, after which the mean and standard deviation are calculated across the independent runs. The evaluation metrics used in the simulation are summarized in Table 4.

Table 4. Evaluation metrics used to assess mission effectiveness, communication resilience, structural robustness, recovery capability, and computational efficiency.

Category	Metric	Definition	Purpose
Mission Performance	Task Completion Rate (TCR)	$\text{TCR} = N_{\text{completed}} / \mathcal{P} $	Measures mission success (S_t , Equation (18)). Higher values indicate better task accomplishment.
	Weighted Task Success (WTS)	$\text{WTS} = \sum_{p \in \mathcal{P}} w_p \mathbb{I}(p \text{ completed}),$	Measures mission effectiveness while accounting for task priorities. Higher values indicate that the coordination strategy successfully completes more important tasks.
	Largest Connected Component Ratio (LCC)	$\text{LCC} = \text{LCC}_t / V_t $	Measures global communication connectivity (C_t , Equation (19)). Higher is better.
	Fragmentation Ratio (FR)	$\text{FR} = (C_t - 1) / (V_t - 1)$	Quantifies network partitioning. Lower values indicate higher resilience.
Structural Resilience	Average Triangle Support (ATS)	$\text{ATS}_t = \frac{1}{ E(B_t) } \sum_{e \in E(B_t)} \text{supp}_{B_t}(e).$	Measures local structural redundancy within the extracted backbone. Higher ATS indicates that retained backbone edges participate in more triangle-supported alternative communication paths.
	Backbone Retention Ratio (BRR)	$\text{BRR}_t = \frac{ E(B_t) }{ E_t^c }$	Measures the proportion of communication edges retained in the extracted k -truss backbone. BRR quantifies backbone coverage and should be interpreted jointly with ATS, since stricter truss constraints may reduce backbone size while increasing structural redundancy.
Efficiency	Energy Consumption (EC)	Total energy consumed during one mission.	Measures the energy term E_t (Equation (21)). Lower is better.
	Execution Time per Coordination Cycle (ET)	Average computation time required to execute one coordination cycle.	Evaluates computational scalability and practical real-time applicability.

Table 5 summarizes the aggregate results over the complete failure sweep. A first observation is that no method dominates all evaluation metrics. Instead, the results reveal a trade-off between mission execution and preservation of global graph properties. This distinction is important for interpreting ATAC because the adaptive k -truss mechanism

is intended to provide a structurally supported substrate for decentralized coordination rather than simply maximize the number or density of communication links.

Table 5. Communication-link failure results (RQ1). Values are mean $\pm$ standard deviation across the five tested link-failure probabilities and 10 independent seeds, computed over the final 25% of each simulation. Arrows indicate the preferred direction. BRR is reported descriptively and is not ranked across methods.

Method	TCR $\uparrow$	LCC $\uparrow$	FR $\downarrow$	ATS $\uparrow$	BRR
GTA	0.839 ± 0.070	0.855 ± 0.121	0.020 ± 0.013	24.141 ± 8.357	1.000 ± 0.001
CPC	0.761 ± 0.073	0.796 ± 0.137	0.022 ± 0.016	21.175 ± 7.393	1.000 ± 0.001
GAC	0.723 ± 0.094	0.865 ± 0.153	0.022 ± 0.016	28.159 ± 8.298	1.000 ± 0.000
ATAC-Fixed	0.879 ± 0.052	0.716 ± 0.158	0.041 ± 0.019	11.276 ± 4.127	0.995 ± 0.004
ATAC	0.871 ± 0.058	0.747 ± 0.134	0.037 ± 0.014	11.940 ± 4.141	0.941 ± 0.041

The task completion results favor the two truss-aware configurations. ATAC-Fixed achieves the highest aggregate TCR, 0.879 ± 0.052 , closely followed by adaptive ATAC at 0.871 ± 0.058 . GTA obtains 0.839 ± 0.070 , while CPC and GAC obtain 0.761 ± 0.073 and 0.723 ± 0.094 , respectively. Relative to the non-truss baselines, adaptive ATAC improves aggregate TCR by approximately 3.8% over GTA, 14.5% over CPC, and 20.5% over GAC. ATAC-Fixed remains slightly higher than adaptive ATAC, with an absolute difference of 0.008, or 0.8 percentage points. This small difference should not be interpreted as evidence that adaptive k universally improves task completion. Rather, it shows that online adjustment of the structural constraint retains mission performance close to the strongest fixed- k configuration.

The behavior of adaptive ATAC across individual failure levels provides a complementary view. Its mean TCR is 0.877, 0.867, 0.863, 0.881, and 0.868 for $p_{\text{fail}} = 0.05, 0.12, 0.18, 0.24, \text{ and } 0.32$, respectively. Thus, increasing the link-failure probability from 0.05 to 0.32 changes mean TCR from 0.877 to 0.868, an absolute reduction of only 0.009. Although the response is not strictly monotonic across the intermediate conditions, task completion remains comparatively stable throughout the tested range.

Figure 9 also resolves an important limitation of the earlier evaluation, in which task completion was saturated and therefore provided little discrimination between methods. Under the regenerated failure conditions, the methods exhibit distinct TCR values, while adaptive ATAC remains close to the strongest fixed- k configuration across the tested failure range.

The graph-level results show a different ordering. GAC obtains the highest aggregate LCC, 0.865 ± 0.153 , followed closely by GTA at 0.855 ± 0.121 . GAC also achieves the highest ATS, 28.159 ± 8.298 , whereas GTA obtains the lowest fragmentation rate, 0.020 ± 0.013 . Adaptive ATAC therefore does not dominate the baselines on every structural metric.

This result is informative rather than contradictory. LCC and ATS describe properties of the communication graph, but neither metric alone measures whether the available communication structure is being used effectively for task execution. GAC, for example, maintains the highest aggregate LCC and ATS while obtaining the lowest aggregate TCR among the five evaluated methods. Conversely, the two ATAC configurations obtain the highest aggregate task completion rates despite operating with lower LCC and ATS values. The results therefore caution against treating raw graph density, component size, or triangle support as standalone surrogates for mission effectiveness.

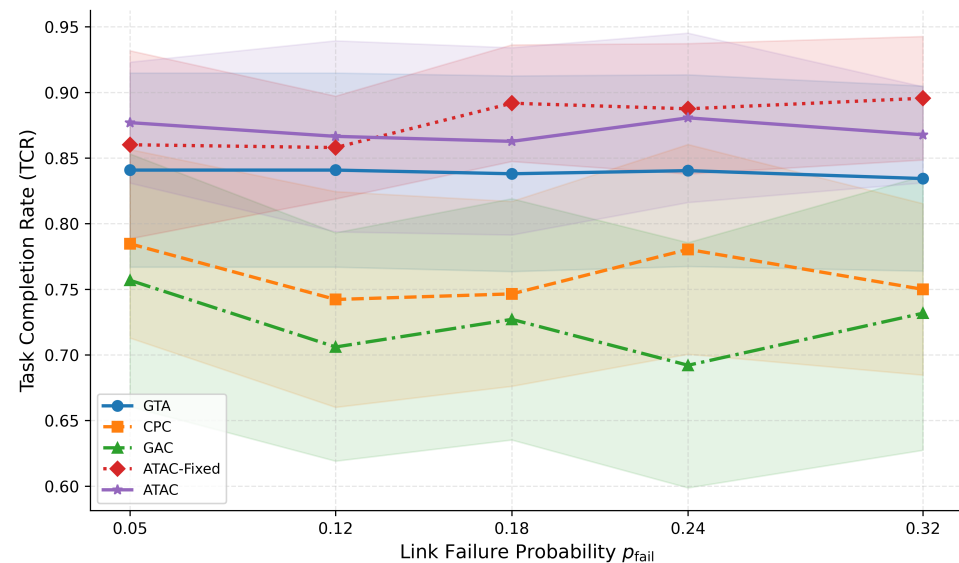


Figure 9. Task-completion rate (TCR) under increasing communication-link failure probability. Curves show the mean over 10 independent seeds, while shaded regions indicate ± 1 standard deviation. Adaptive ATAC maintains comparatively stable task completion across the tested communication-degradation conditions.

Figures 10–12 make this distinction visible across the complete failure sweep. The methods that preserve the largest or most triangle-supported communication graph are not necessarily those that achieve the highest task completion rate. This supports evaluating mission-level and structural metrics jointly rather than interpreting any single graph statistic as a complete measure of resilience.

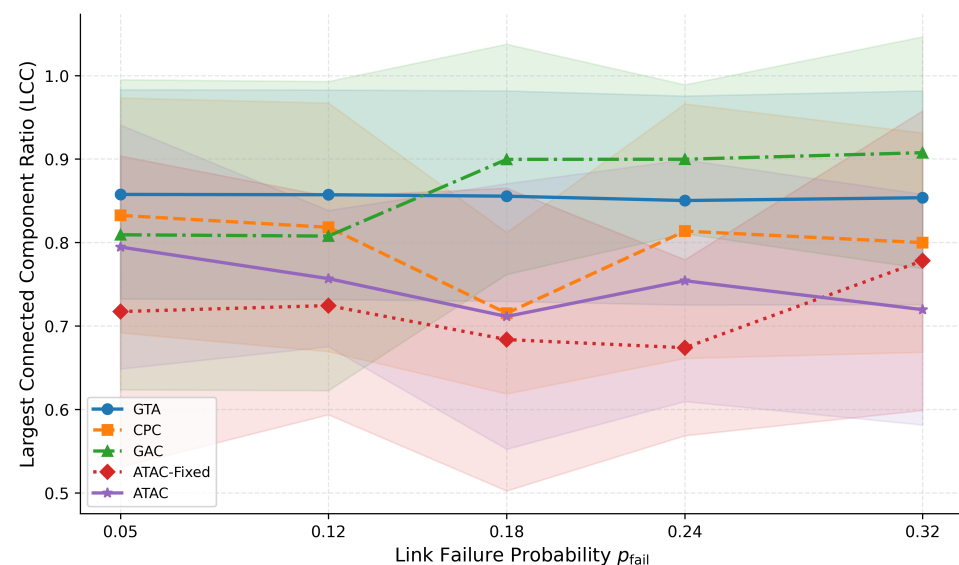


Figure 10. Largest connected component ratio (LCC) under increasing communication-link failure probability. Curves show the mean over 10 independent seeds, while shaded regions indicate ± 1 standard deviation.

The comparison between ATAC-Fixed and adaptive ATAC provides a more direct assessment of the adaptive mechanism. Relative to ATAC-Fixed, adaptive ATAC increases aggregate LCC from 0.716 to 0.747 and ATS from 11.276 to 11.940 while reducing fragmentation from 0.041 to 0.037. These changes correspond to an approximately 4.3% increase in LCC, 5.9% increase in ATS, and 9.8% reduction in fragmentation, while aggregate TCR

decreases by only 0.008. These differences are descriptive and are not claimed to be statistically significant in the absence of a dedicated paired significance test. Nevertheless, they show that adaptive k changes the operating point of the truss-aware policy rather than merely reproducing the behavior of the fixed- k configuration.

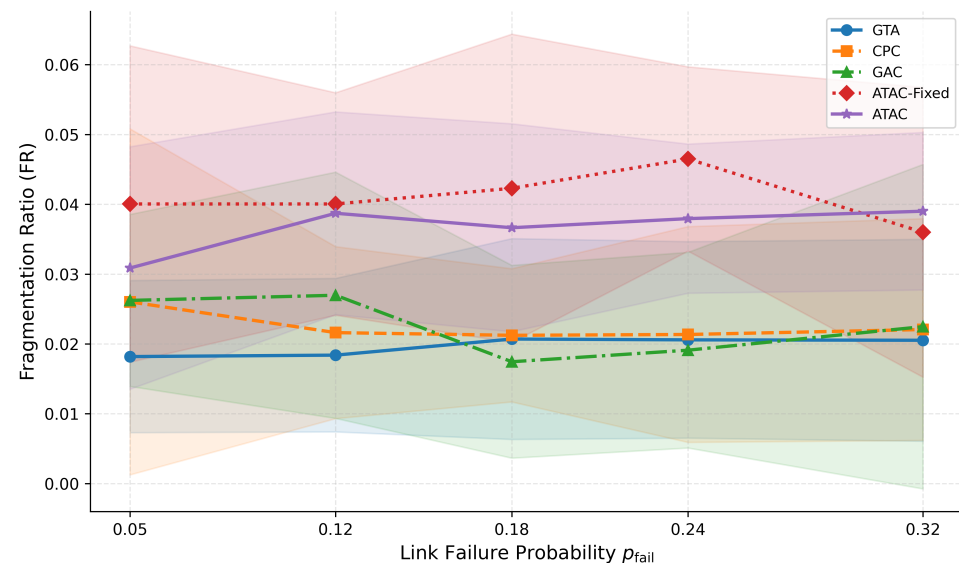


Figure 11. Fragmentation ratio (FR) under increasing communication-link failure probability. Curves show the mean over 10 independent seeds, while shaded regions indicate ± 1 standard deviation. Lower values indicate less fragmentation of the communication graph.

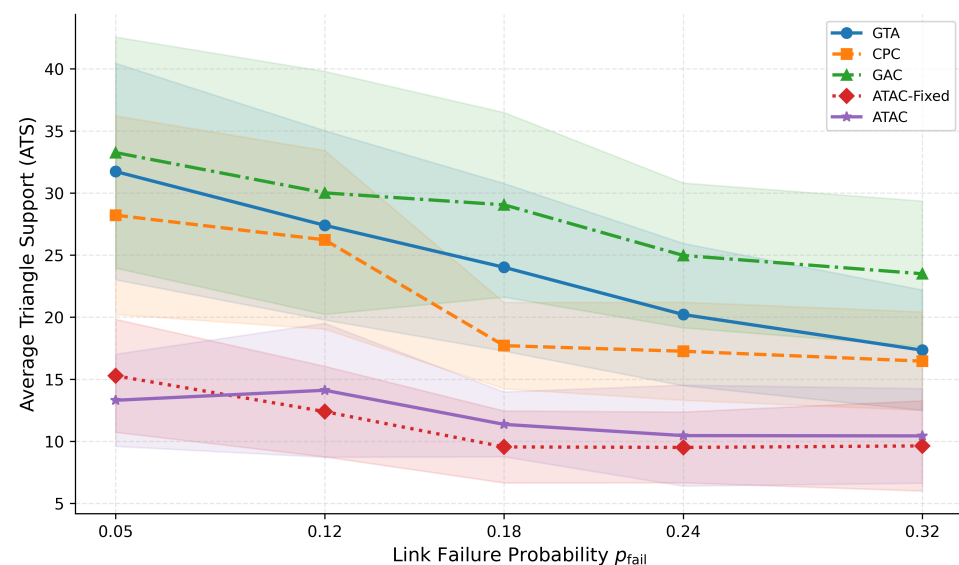


Figure 12. Average triangle support (ATS) under increasing communication-link failure probability. Curves show the mean over 10 independent seeds, while shaded regions indicate ± 1 standard deviation.

The BRR values require additional care in interpretation. GTA, CPC, and GAC exhibit BRR values close to unity, while adaptive ATAC obtains 0.941 ± 0.041 . BRR should not, however, be interpreted as a direct cross-method performance ranking. For the truss-aware policies, the backbone is deliberately a selective subgraph of the instantaneous communication graph, and adaptive selection of k can remove edges that do not satisfy the selected structural requirement. A smaller retained fraction can therefore reflect a more selective backbone rather than simply a loss of communication capability. For this reason,

BRR is reported descriptively in Table 5 and is interpreted together with task completion, connectivity, fragmentation, and triangle support.

Most importantly, the selected truss level changes systematically as communication conditions deteriorate. At $p_{\text{fail}} = 0.05$, ATAC selects $k = 5$ throughout the observed trajectory. At $p_{\text{fail}} = 0.12$, approximately 98.7% of the selections remain at $k = 5$. This proportion decreases to approximately 94.3% at $p_{\text{fail}} = 0.18$ and 86.0% at $p_{\text{fail}} = 0.24$. Under the most severe tested condition, $p_{\text{fail}} = 0.32$, approximately 73.4% of the selections use $k = 5$, 23.4% use $k = 4$, and 3.2% use $k = 3$. Correspondingly, the mean selected truss level decreases from approximately 5.00 at the lowest failure probability to approximately 4.70 at the highest.

Figure 13 makes the adaptation mechanism explicit. The distribution of the selected truss level progressively shifts toward lower values as communication degradation increases, rather than remaining concentrated at one fixed value of k . The direction of this change is consistent with the intended role of adaptive truss selection. When sufficient triangle-supported connectivity is available, ATAC can maintain a stronger structural requirement. As communication links are lost and stronger requirements become more difficult to sustain, lower values of k are selected more frequently.

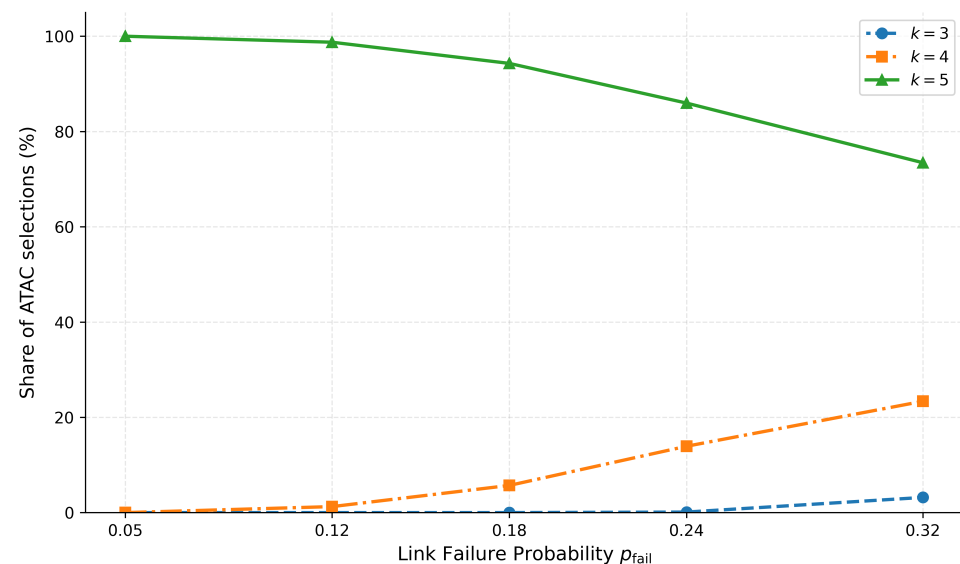


Figure 13. Adaptive truss-level selection under increasing communication-link failure probability. Lower truss levels are selected increasingly often as communication degradation becomes more severe. At $p_{\text{fail}} = 0.32$, approximately 73.4% of the selections use $k = 5$, 23.4% use $k = 4$, and 3.2% use $k = 3$.

The observed change therefore provides direct evidence that the adaptive mechanism responds systematically to changes in the evolving communication topology rather than behaving as an effectively fixed truss policy. Importantly, this interpretation does not require adaptive ATAC to maximize every individual graph metric. Its role is to modify the structural constraint online as the conditions under which coordination takes place change.

This behavior is particularly relevant to the disaster-response setting that motivates the framework. During a mission, the communication structure available to a UAV swarm may change because of mobility, obstruction, interference, or infrastructure disruption. A coordination strategy that assumes that the same structural requirement remains appropriate throughout the mission may therefore become unnecessarily restrictive as conditions deteriorate. Within the controlled simulation considered here, adaptive ATAC instead modifies the truss requirement as the available topology changes.

Taken together, the RQ1 results reveal a more informative resilience trade-off than a single-metric ranking. Adaptive ATAC achieves the second-highest aggregate TCR (0.871 ± 0.058), only 0.008 below ATAC-Fixed, while outperforming GTA, CPC, and GAC in aggregate task completion. Relative to ATAC-Fixed, adaptation also improves LCC and ATS and reduces fragmentation. At the same time, the selected truss level changes systematically as communication conditions deteriorate, with lower values of k occurring increasingly often at higher link-failure probabilities.

RQ1 therefore supports the intended role of adaptive k : not to maximize every graph metric independently, but to adjust the structural constraint as communication conditions change while retaining strong mission-level performance. Within the tested conditions, ATAC provides a mechanism for changing the balance between task execution and structural connectivity rather than assuming that a single fixed connectivity requirement remains appropriate throughout the mission.

7.5. RQ2: Resilience Under UAV Node Failures

RQ2 examines how the coordination strategies respond when individual UAVs become unavailable during an ongoing mission. Unlike RQ1, where the UAVs remain operational but communication links are intermittently removed, node failures directly reduce the number of agents available for sensing, task execution, and communication. This provides a complementary resilience test that is particularly relevant to disaster-response operations, where UAVs may become unavailable because of battery depletion, hardware faults, collision, environmental exposure, emergency landing, or loss of control.

We consider four node-failure probabilities,

$$p_{\text{node}} \in \{0, 0.0008, 0.0016, 0.0028\},$$

and evaluate GTA, CPC, GAC, ATAC-Fixed, and adaptive ATAC. Each method–failure combination is evaluated using 10 independent random seeds and 260 coordination steps. As in RQ1, the reported statistics are computed over the final 25% of each simulation. For each seed, a metric is first averaged over this late-mission interval, after which the mean and standard deviation are calculated across the independent runs.

Table 6 summarizes the aggregate results over the complete node-failure sweep. As in the communication-link failure experiment, no single method dominates every metric. Instead, the results distinguish between mission completion, preservation of global graph properties, and resource expenditure.

Table 6. UAV node-failure results (RQ2). Values are mean $\pm$ standard deviation across the four tested node-failure probabilities and 10 independent seeds, computed over the final 25% of each simulation. Arrows indicate the preferred direction.

Method	TCR $\uparrow$	WTS $\uparrow$	LCC $\uparrow$	FR $\downarrow$	ATS $\uparrow$	Energy $\downarrow$
GTA	0.843 ± 0.063	0.886 ± 0.051	0.864 ± 0.139	0.027 ± 0.025	18.886 ± 7.467	21.361 ± 3.475
CPC	0.738 ± 0.096	0.795 ± 0.083	0.764 ± 0.142	0.040 ± 0.028	15.624 ± 5.586	14.821 ± 2.085
GAC	0.737 ± 0.091	0.797 ± 0.084	0.835 ± 0.143	0.030 ± 0.021	21.080 ± 8.832	15.917 ± 2.502
ATAC-Fixed	0.868 ± 0.058	0.875 ± 0.073	0.686 ± 0.131	0.059 ± 0.022	7.510 ± 3.531	5.631 ± 0.601
ATAC	0.857 ± 0.059	0.857 ± 0.075	0.729 ± 0.148	0.049 ± 0.022	9.391 ± 3.347	5.770 ± 0.442

The mission-level results favor the two truss-aware configurations. ATAC-Fixed achieves the highest aggregate TCR, 0.868 ± 0.058 , followed closely by adaptive ATAC at 0.857 ± 0.059 . GTA obtains 0.843 ± 0.063 , whereas CPC and GAC obtain 0.738 ± 0.096 and 0.737 ± 0.091 , respectively. Relative to the non-truss baselines, adaptive ATAC improves aggregate TCR by approximately 1.7% over GTA, 16.1% over CPC, and 16.3% over GAC.

The difference between ATAC and ATAC-Fixed is only 0.011 in absolute TCR. Accordingly, these results do not suggest that adaptation simply maximizes task completion; rather, they indicate that adaptive ATAC retains mission performance close to the strongest fixed configuration while modifying the structural operating point as UAVs are lost. Figure 14 illustrates the TCR trends across the tested UAV node-failure probabilities.

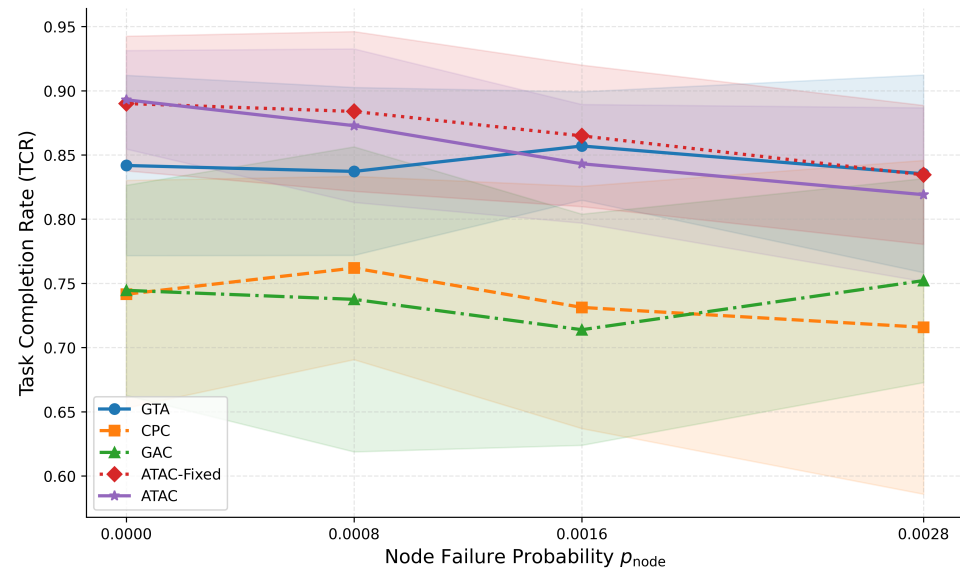


Figure 14. Task- completion rate (TCR) under increasing UAV node-failure probability. Curves show the mean over 10 independent seeds, while shaded regions indicate ± 1 standard deviation.

The graph-level results show a different ordering. GTA achieves the highest aggregate LCC (0.864 ± 0.139) and the lowest fragmentation rate (0.027 ± 0.025), while GAC achieves the highest ATS (21.080 ± 8.832). Adaptive ATAC therefore does not maximize every global graph statistic. This mirrors the result observed in RQ1: preserving the largest or most triangle-rich full communication graph is not equivalent to maximizing task completion. In particular, GAC exhibits substantially higher ATS than adaptive ATAC but a considerably lower aggregate TCR (0.737 versus 0.857).

The comparison between adaptive ATAC and ATAC-Fixed provides the most direct evidence about the effect of adaptive truss selection. Relative to the fixed configuration, adaptive ATAC increases aggregate LCC from 0.686 to 0.729 and ATS from 7.510 to 9.391, while reducing fragmentation from 0.059 to 0.049. These correspond to an approximately 6.3% increase in LCC, 25.0% increase in ATS, and 16.9% reduction in fragmentation. These structural changes are obtained while aggregate TCR decreases by only 0.011.

The difference becomes particularly visible at the most severe tested node-failure condition, $p_{\text{node}} = 0.0028$. Under this condition, adaptive ATAC obtains an LCC of approximately 0.759, compared with approximately 0.660 for ATAC-Fixed, corresponding to an increase of about 15.0%. Fragmentation decreases from approximately 0.0817 to 0.0567, a reduction of about 30.6%, while ATS increases from approximately 4.521 to 6.917, or about 53.0%. These differences are reported descriptively and are not claimed to be statistically significant in the absence of a dedicated paired significance test. They nevertheless show that the effect of adaptation becomes more pronounced when node degradation is strongest. The corresponding LCC and fragmentation trends across the tested node-failure probabilities are shown in Figures 15 and 16, respectively.

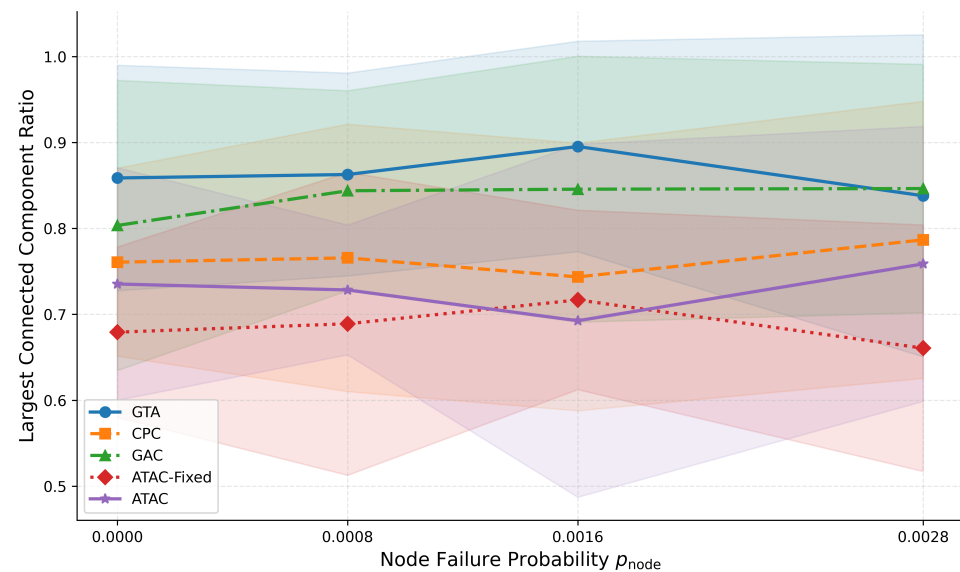


Figure 15. Largest connected component ratio (LCC) under increasing UAV node-failure probability. Curves show the mean over 10 independent seeds, while shaded regions indicate ± 1 standard deviation.

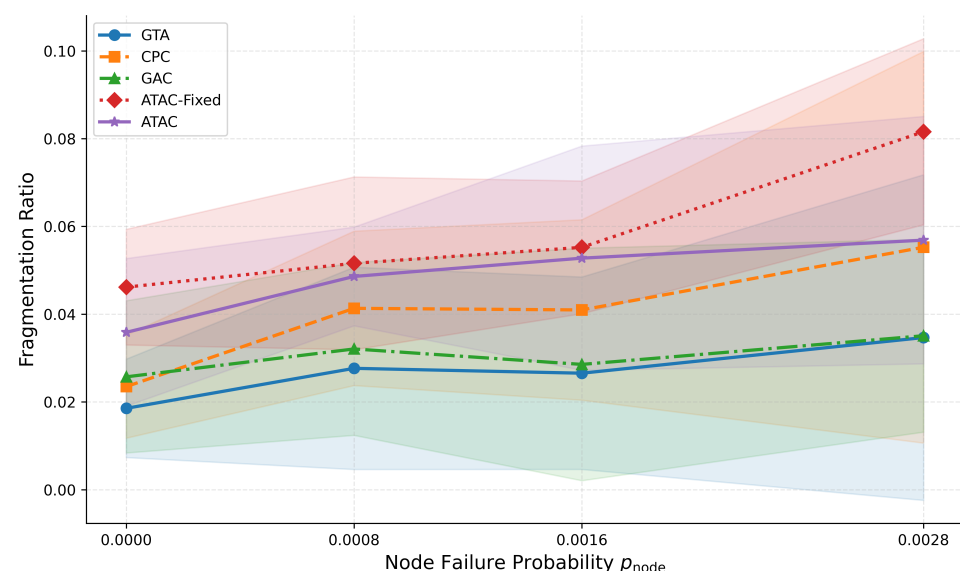


Figure 16. Fragmentation ratio (FR) under increasing UAV node-failure probability. Curves show the mean over 10 independent seeds, while shaded regions indicate ± 1 standard deviation. Lower values indicate less fragmentation.

Resource expenditure provides another important dimension of the comparison. Adaptive ATAC uses 5.770 ± 0.442 simulator energy units, compared with 21.361 ± 3.475 for GTA, 14.821 ± 2.085 for CPC, and 15.917 ± 2.502 for GAC. Relative to these non-truss baselines, this corresponds to reductions in modeled energy expenditure of approximately 73.0%, 61.1%, and 63.8%, respectively. ATAC-Fixed is slightly lower at 5.631 ± 0.601 . These values are specific to the energy model used in the simulator and should therefore not be interpreted directly as physical flight time or battery endurance. Within the common simulation model, however, the results show that the high task completion performance of the truss-aware strategies is not obtained through greater modeled energy expenditure.

A central question in RQ2 is whether adaptive k actually responds to the progressive loss of UAVs. The observed selection frequencies show a systematic change. With no node failures, $p_{\text{node}} = 0$, approximately 94.9% of ATAC selections use $k = 5$ and 5.1% use $k = 4$.

At $p_{\text{node}} = 0.0008$, the corresponding proportions are approximately 94.0% and 6.0%. At $p_{\text{node}} = 0.0016$, the share of $k = 5$ decreases to approximately 91.5%, while $k = 4$ increases to 8.3% and $k = 3$ appears in approximately 0.1% of the selections.

The shift becomes clearer under the strongest tested node degradation. At $p_{\text{node}} = 0.0028$, the share of $k = 5$ decreases to approximately 80.8%, while $k = 4$ rises to 18.7% and $k = 3$ accounts for approximately 0.5% of the selections.

Figure 17 makes this response explicit. The adaptive mechanism does not remain concentrated at a single value of k as UAV losses increase. Instead, stronger node degradation is associated with progressively more frequent selection of lower truss levels. This result complements RQ1, where the same qualitative shift was observed under communication-link failures. The two experiments therefore show that adaptive k responds systematically under two distinct forms of degradation: loss of communication links and loss of UAV nodes.

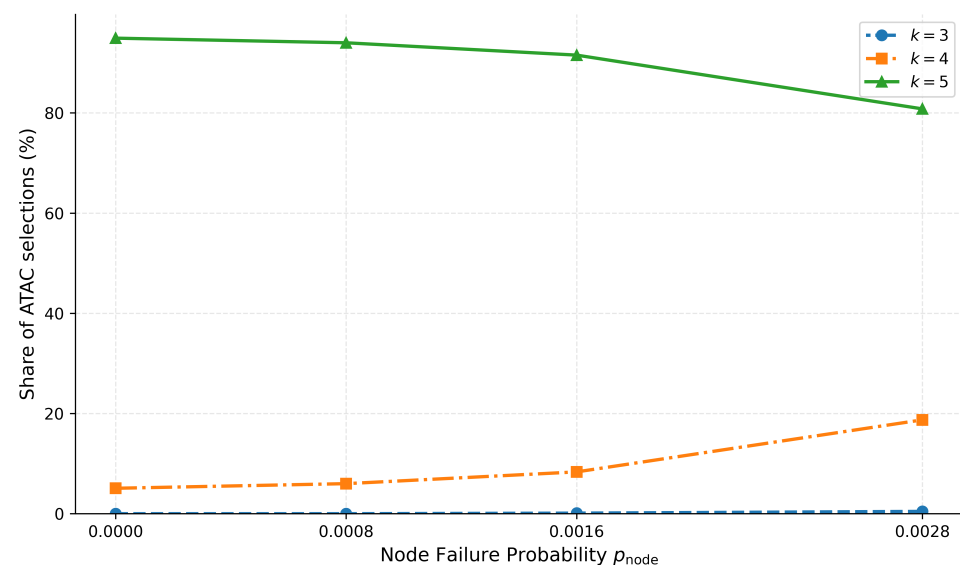


Figure 17. Distribution of the truss level selected by adaptive ATAC under increasing UAV node-failure probability. As node degradation increases, the share of $k = 5$ decreases and lower truss levels are selected more frequently, showing that the structural requirement is adjusted as the surviving communication topology changes.

This behavior has a direct interpretation in the disaster-response setting considered in this work. When a UAV becomes unavailable, the effect is not limited to the loss of one sensing or task-execution resource; its incident communication links are also removed, changing the topology available to the remaining swarm. Maintaining an unchanged structural requirement may therefore become increasingly restrictive as additional UAVs are lost. Adaptive ATAC instead allows the required truss level to relax when the surviving topology provides less structural support. The node-failure experiment should be viewed as a controlled abstraction of this resource degradation rather than as a physical model of any specific UAV failure mechanism.

Taken together, RQ2 shows that adaptive ATAC maintains strong mission-level performance under progressive UAV loss while changing its structural operating point as the surviving network deteriorates. Adaptive ATAC achieves the second-highest aggregate TCR (0.857 ± 0.059), only 0.011 below ATAC-Fixed, and outperforms the three non-truss baselines in aggregate task completion. Relative to ATAC-Fixed, it increases LCC and ATS and reduces fragmentation, with the largest descriptive differences occurring at the highest tested node-failure probability. It also retains substantially lower modeled energy expenditure than GTA, CPC, and GAC.

RQ2 therefore supports the intended role of adaptive k as a mechanism for adjusting the structural requirement when the set of available UAVs changes during a mission. Together with RQ1, the results indicate that ATAC does not require one fixed structural operating point across different degradation conditions. Instead, it can relax the truss requirement as either communication links or UAV nodes are lost while retaining high task completion performance.

7.6. RQ3: Sensitivity to the Truss Level

RQ3 examines how the choice of the truss level k affects ATAC and whether a single fixed value provides a consistently preferable operating point. We compare fixed configurations with $k \in \{3, 4, 5\}$ against adaptive ATAC under the same communication condition, $p_{\text{fail}} = 0.18$, with no node failures. Each configuration is evaluated with 10 independent seeds and 260 coordination steps. As in RQ1 and RQ2, the reported statistics are computed over the late-mission evaluation window.

Table 7 shows that the choice of k materially affects the operating point. ATAC- $k4$ achieves the highest TCR (0.911 ± 0.049) and the lowest fragmentation (0.031 ± 0.016), whereas ATAC- $k5$ provides the highest LCC (0.786 ± 0.147) and ATS (12.542 ± 3.370). ATAC- $k3$, in turn, has the lowest modeled energy expenditure. Thus, no fixed value is uniformly preferable across all of the reported criteria.

Table 7. Sensitivity to fixed and adaptive truss-level selection (RQ3) under $p_{\text{fail}} = 0.18$. Values are mean $\pm$ standard deviation across 10 independent seeds over the late-mission evaluation window. Arrows indicate the preferred direction. BRR is reported descriptively because it depends directly on the selected truss requirement.

Configuration	TCR $\uparrow$	LCC $\uparrow$	FR $\downarrow$	ATS $\uparrow$	BRR	Energy $\downarrow$
ATAC- $k3$	0.892 ± 0.044	0.691 ± 0.176	0.042 ± 0.022	9.350 ± 2.849	0.996 ± 0.003	5.448 ± 0.497
ATAC- $k4$	0.911 ± 0.049	0.761 ± 0.158	0.031 ± 0.016	11.373 ± 2.981	0.982 ± 0.008	5.523 ± 0.402
ATAC- $k5$	0.876 ± 0.065	0.786 ± 0.147	0.032 ± 0.014	12.542 ± 3.370	0.953 ± 0.015	5.949 ± 0.736
ATAC-Adaptive	0.862 ± 0.074	0.715 ± 0.157	0.036 ± 0.014	11.330 ± 2.696	0.939 ± 0.019	5.937 ± 0.523

The structural effect of k is also substantial. Increasing k from 3 to 5 raises LCC from 0.691 to 0.786 and ATS from 9.350 to 12.542, corresponding to improvements of approximately 13.7% and 34.1%, respectively. However, the highest TCR occurs at $k = 4$, rather than at the strongest tested truss level. This illustrates why k should be viewed as a structural control parameter rather than as a neutral implementation constant.

Figure 18 makes this trade-off more apparent. A stronger truss requirement improves some structural properties but does not necessarily improve task completion. This is the main motivation for allowing the structural requirement to change when the network conditions themselves change.

Under the stationary condition used in RQ3, adaptive ATAC selects $k = 5$ for approximately 94.1% of all trajectory observations and $k = 4$ for the remaining 5.9%. During the late-mission evaluation window, all adaptive runs settle at $k = 5$.

The behavior in Figure 19 is useful when read together with RQ1 and RQ2. Adaptation does not imply that k must continually change. When the topology continues to support a stronger truss requirement, the controller can remain at that level. In contrast, RQ1 and RQ2 showed that lower values of k are selected more frequently as link or node degradation becomes more severe.

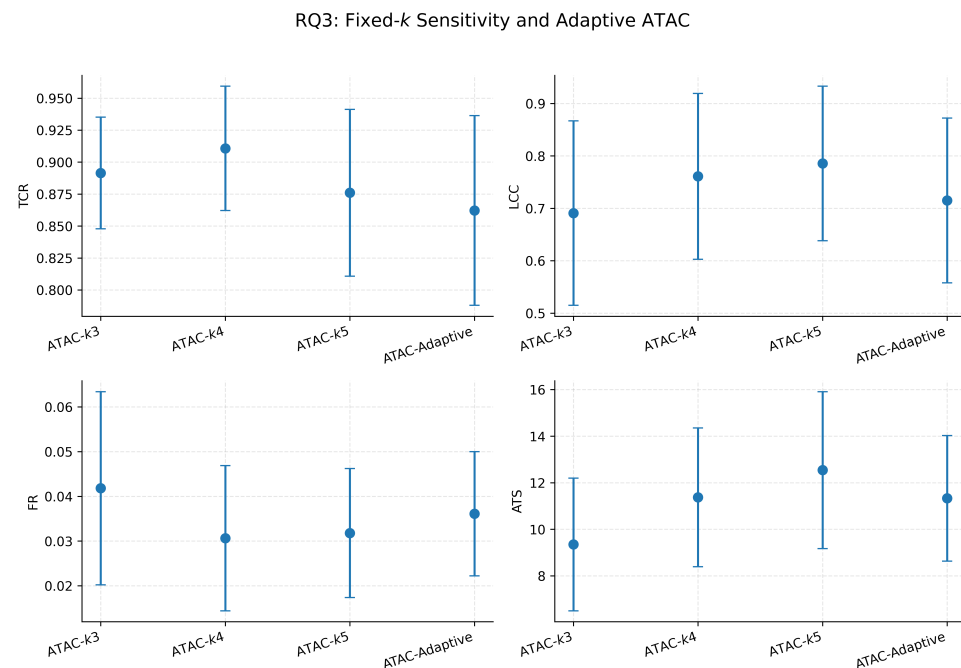


Figure 18. Sensitivity of ATAC to the truss level under $p_{\text{fail}} = 0.18$. Points show the mean across 10 independent seeds over the late-mission evaluation window, with error bars indicating ± 1 standard deviation. Different values of k favor different mission and structural metrics.

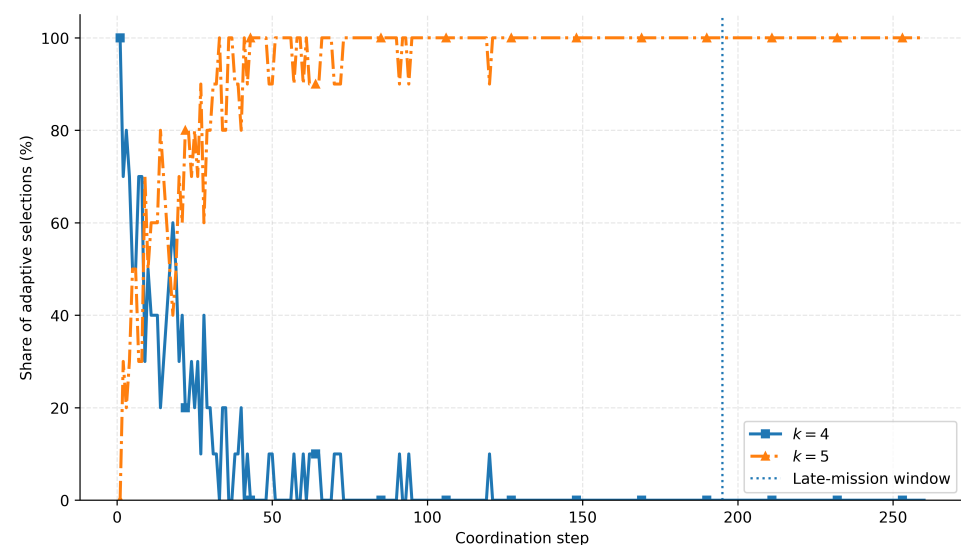


Figure 19. Evolution of the truss level selected by adaptive ATAC in RQ3. The curves show the proportion of the 10 runs selecting each value of k at every coordination step. Under this stationary condition, the controller progressively settles on $k = 5$.

RQ3 therefore does not show that adaptive ATAC always outperforms the best fixed k under a stationary condition; indeed, $k = 4$ performs better on TCR in this simulation study. Rather, it shows that different values of k lead to different mission–structure trade-offs. Together with the adaptive behavior observed under the changing conditions of RQ1 and RQ2, this supports using k as an online structural control variable instead of prescribing one fixed value for the entire mission.

7.7. RQ4: Ablation Analysis

RQ4 examines how the main components of the proposed framework affect mission and structural performance. We compare GTA, GAC, ATAC-Fixed, and the complete

adaptive ATAC configuration under the same communication condition, $p_{\text{fail}} = 0.24$, with $N = 50$ UAVs and no node failures. Each configuration is evaluated using 12 independent seeds and 280 coordination steps. The reported values are computed over the late-mission evaluation window.

Table 8 shows that the different components produce distinct operating points rather than a single uniformly dominant configuration. ATAC-Fixed achieves the highest aggregate TCR (0.887 ± 0.062), while adaptive ATAC remains close at 0.869 ± 0.033 . Both truss-aware configurations outperform GTA and GAC in task completion under this condition.

Table 8. Ablation results for RQ4. Values are mean $\pm$ standard deviation across 12 independent seeds over the late-mission evaluation window. Arrows indicate the preferred direction.

Method	TCR $\uparrow$	LCC $\uparrow$	FR $\downarrow$	ATS $\uparrow$	Energy $\downarrow$	RT/Step (s) $\downarrow$
GTA	0.808 ± 0.071	0.848 ± 0.094	0.025 ± 0.015	21.055 ± 3.956	25.454 ± 2.218	0.971 ± 0.195
GAC	0.705 ± 0.080	0.900 ± 0.152	0.021 ± 0.016	28.579 ± 7.564	21.048 ± 2.182	1.322 ± 0.310
ATAC-Fixed	0.887 ± 0.062	0.658 ± 0.135	0.047 ± 0.016	8.648 ± 2.196	6.254 ± 0.516	0.532 ± 0.139
ATAC	0.869 ± 0.033	0.739 ± 0.199	0.037 ± 0.013	10.157 ± 2.916	6.552 ± 0.522	0.624 ± 0.125

The comparison between ATAC and ATAC-Fixed is particularly informative. Adaptive selection increases LCC from 0.658 to 0.739, corresponding to an improvement of approximately 12.3%. It also reduces fragmentation from 0.047 to 0.037, a reduction of approximately 21.4%, and increases ATS from 8.648 to 10.157, or about 17.5%. These structural gains are obtained with a relatively small reduction in aggregate TCR of about 2.0% relative to ATAC-Fixed.

Figure 20 highlights the same trade-off. GAC preserves the strongest global graph structure, whereas the two truss-aware variants achieve higher task completion. Adaptive ATAC occupies an intermediate structural operating point, improving LCC, fragmentation, and ATS relative to ATAC-Fixed without a large loss in mission completion.

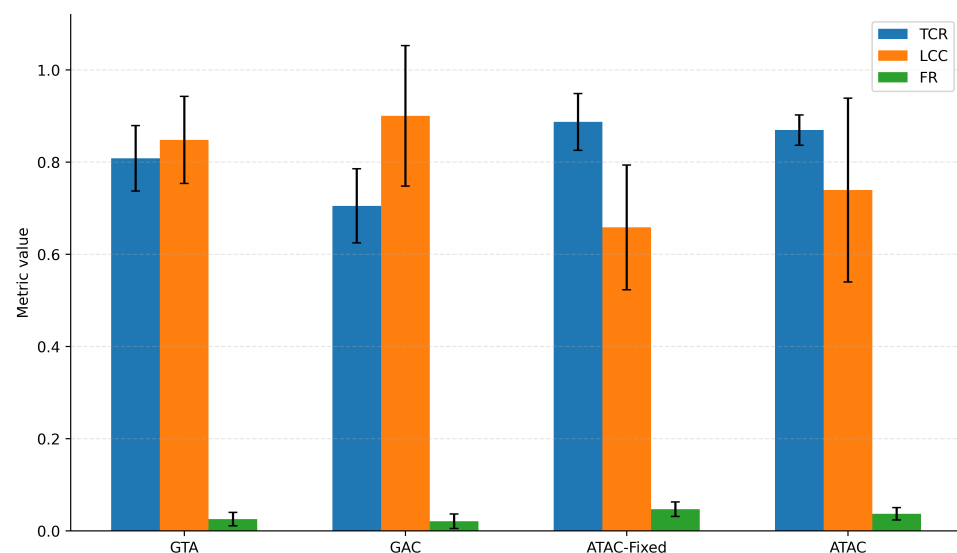


Figure 20. Ablation comparison of GTA, GAC, ATAC-Fixed, and adaptive ATAC under $p_{\text{fail}} = 0.24$. Bars show mean TCR, LCC, and fragmentation rate across 12 independent seeds; error bars indicate ± 1 standard deviation.

The resource results also support the usefulness of the truss-aware design. Adaptive ATAC uses 6.552 ± 0.522 modeled energy units, compared with 25.454 ± 2.218 for GTA and 21.048 ± 2.182 for GAC. This corresponds to reductions of approximately 74.3% and 68.9%,

respectively. Its runtime per coordination step is also lower than GTA and GAC, although ATAC-Fixed remains the fastest of the two truss-aware variants.

Overall, RQ4 shows that the complete adaptive mechanism does not simply maximize one metric. Instead, it modifies the balance between mission completion, structural cohesion, and resource cost. Relative to the fixed- k configuration, adaptive ATAC sacrifices only a small amount of TCR while improving all three reported structural indicators. This supports the role of adaptive truss selection as a mechanism for changing the mission–structure operating point when communication conditions evolve.

7.8. RQ5: Scalability with Swarm Size

RQ5 evaluates how ATAC behaves as the swarm size increases. We consider $N \in \{20, 32, 50, 70, 90\}$ UAVs under the same communication condition, $p_{\text{fail}} = 0.17$, with no node failures. ATAC is compared with GAC, the graph-aware baseline, using six independent seeds for each swarm size. Each run contains 190 coordination steps, and the reported performance statistics are computed over the final 25% of the run.

Table 9 shows that ATAC remains operational across the full tested range, but the effect of swarm size is not uniform across all metrics. ATAC achieves a higher TCR than GAC for $N = 20, 32$, and 50 , reaching its highest value at $N = 50$ (0.837). At $N = 70$ and 90 , GAC obtains the higher TCR. This suggests that increasing the number of UAVs beyond a certain point does not automatically translate into higher task completion for ATAC.

Table 9. Scalability results for GAC and ATAC. Values are means across six independent seeds over the late-mission evaluation window. $\uparrow$ indicates that a higher value is preferred, whereas $\downarrow$ indicates that a lower value is preferred.

N	Method	TCR $\uparrow$	LCC $\uparrow$	FR $\downarrow$	ATS $\uparrow$	Energy $\downarrow$	RT/Step (s) $\downarrow$
20	GAC	0.514	0.686	0.105	8.662	15.022	0.126
	ATAC	0.634	0.509	0.132	3.151	11.332	0.069
32	GAC	0.653	0.683	0.043	13.997	14.707	0.311
	ATAC	0.666	0.747	0.055	7.303	8.708	0.214
50	GAC	0.802	0.694	0.031	21.972	14.468	0.809
	ATAC	0.837	0.733	0.039	9.191	5.896	0.504
70	GAC	0.799	0.760	0.021	31.169	14.421	2.095
	ATAC	0.785	0.851	0.015	11.681	4.881	1.036
90	GAC	0.841	0.647	0.024	34.838	12.912	2.712
	ATAC	0.781	0.926	0.009	14.946	4.733	1.893

The structural results show a different trend. ATAC's LCC increases from 0.509 at $N = 20$ to 0.926 at $N = 90$, while its fragmentation rate falls from 0.132 to 0.009. At $N = 90$, ATAC therefore maintains a substantially larger connected component than GAC (0.926 versus 0.647) and less fragmentation (0.009 versus 0.024). The larger swarm provides more opportunities for the truss-aware mechanism to form triangle-supported communication structures.

Figure 21 also illustrates that scalability should not be interpreted only in terms of task completion. ATAC performs well at small and medium swarm sizes, but the TCR advantage over GAC does not persist at the two largest tested configurations. This result indicates a trade-off between mission execution and the increasingly structured coordination maintained by ATAC.

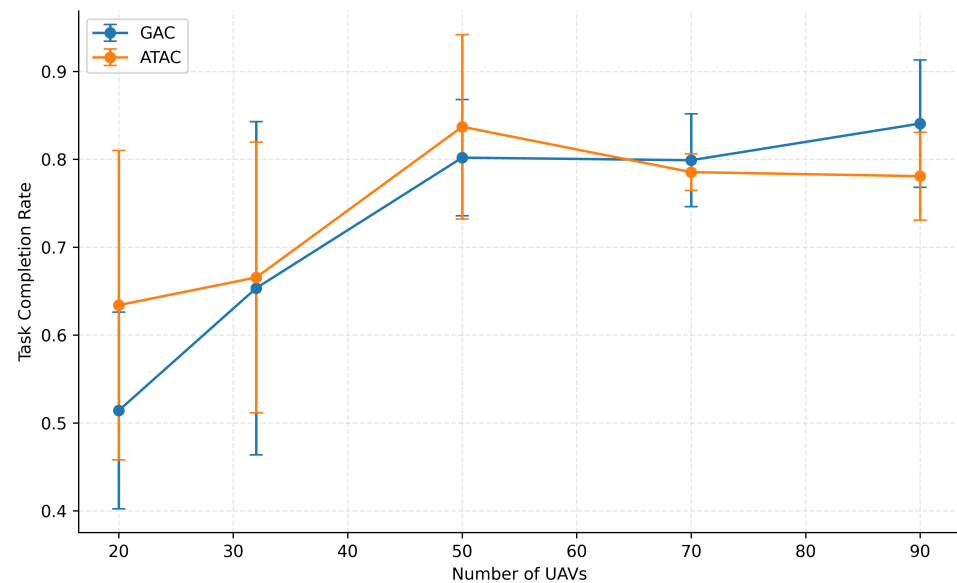


Figure 21. Task completion rate as swarm size increases. Points show the mean across six independent seeds and error bars indicate ± 1 standard deviation.

As expected, the computational cost increases with swarm size (Figure 22). For ATAC, the mean runtime per coordination step increases from 0.069 s at $N = 20$ to 1.893 s at $N = 90$. GAC increases from 0.126 s to 2.712 s over the same range. Thus, although neither method exhibits constant computational cost, ATAC remains faster than GAC at every swarm size evaluated.

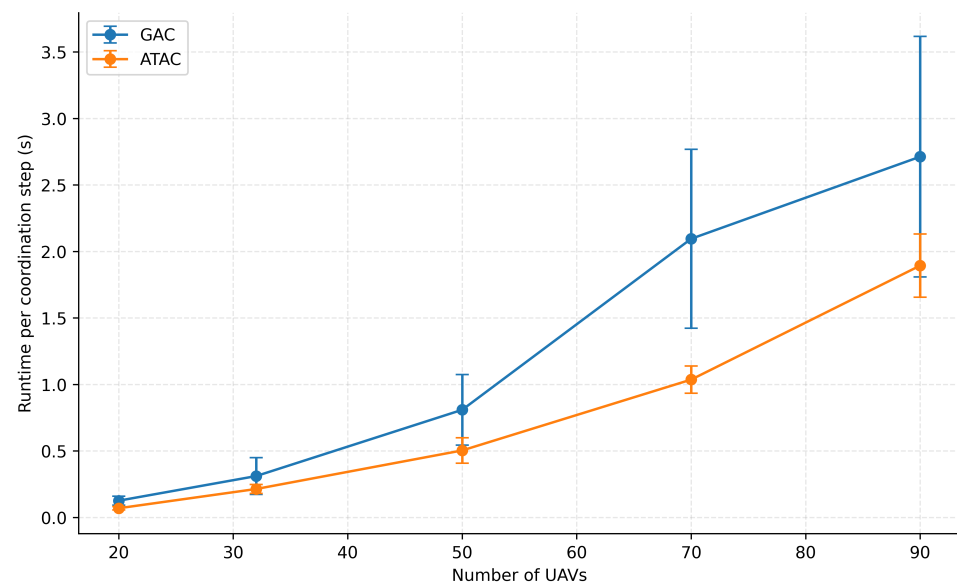


Figure 22. Runtime per coordination step as a function of swarm size for GAC and ATAC. Points show the mean across six independent seeds and error bars indicate ± 1 standard deviation.

The energy results follow a similarly favorable trend for ATAC. At $N = 90$, its modeled energy expenditure is 4.733 units, compared with 12.912 for GAC, a reduction of approximately 63.3%. As in the previous experiments, these values are simulator-specific and should not be interpreted directly as physical flight endurance.

Overall, RQ5 shows that ATAC can be executed for swarms of up to 90 UAVs in the tested simulation while maintaining strong structural connectivity and lower computational and modeled energy costs than GAC. The results do not establish scalability

beyond this range, nor do they show that increasing swarm size always improves task completion. Instead, they define the tested operating range and show that ATAC remains computationally tractable while preserving its structural coordination properties as the swarm grows.

7.9. RQ6: Reward-Weight Sensitivity Analysis

The final simulation study evaluates the sensitivity of ATAC to the mission-level reward weights used by the graph-aware decision policy. The goal is to understand if the framework favors a specific configuration and how it balances the trade-offs between the considered mission objectives. The five reward profiles include balanced, task-oriented, connectivity-oriented, resilience-oriented, and energy-aware. The balanced configuration was used as a reference, where all four mission objectives (task completion, connectivity, structural resilience, and energy) were given equal weights. The other profiles had higher weights on a specific objective compared to the balanced configuration. All configurations were tested under three disaster-intensity levels: mild, moderate, and severe. The latter were implemented by varying the probabilities of communication-link and UAV-node failures. Each configuration was evaluated using the same simulation environment and independent random seeds.

Table 10 shows that varying the reward weights leads to distinct operating behaviors of ATAC. In particular, the task-oriented configuration had the best mission performance, yielding the highest mean task completion rate (TCR) of 0.879 and weighted task success (WTS) of 0.864. Surprisingly, it also had the highest mean largest connected component ratio (LCC) of 0.818 and the lowest fragmentation ratio (FR) of 0.040 among all configurations. This suggests that, within the tested range of weights, prioritizing the task execution did not hurt the global connectivity performance. The connectivity-oriented configuration had a similar performance in terms of LCC (0.810) and FR (0.042) but a lower TCR (0.781) than the task-oriented configuration. This implies that an extreme emphasis on connectivity can limit the task progress by constraining the UAV motion. In turn, the energy-aware configuration was able to reduce the energy consumption while having a moderate impact on the mission performance. In particular, it had the lowest mean energy consumption of 4.060, a TCR of 0.851, and a WTS of 0.858. This suggests that energy-aware moderation can be an effective way to limit the mobility costs without introducing large degradation in the mission performance. Finally, the resilience-oriented configuration was not superior in any of the considered performance metrics. Its mean TCR, LCC, and FR were 0.776, 0.648, and 0.063, respectively. This indicates that, within the tested range of weights, an increased emphasis on the structural resilience did not yield any significant benefits in terms of mission performance. This result is particularly interesting because it shows that the reward weights shaping the structural term should not be increased independently of other terms. In particular, an extreme emphasis on structural resilience can limit the task progress and disrupt the global connectivity, which ultimately degrades the LCC and increases the FR. Thus, the reward weights shaping different aspects of the UAV motion should be viewed as interacting control variables. Figure 23 provides additional insight into the reward-weight sensitivity by showing how the performance metrics vary with the disaster intensity. In particular, while the performance of most configurations was lower under higher disaster intensity, the relative trends remained similar. For example, the task-oriented configuration still had the best TCR across all disaster levels. At the same time, the energy-aware configuration provided the best energy-performance trade-off under moderate and severe disaster conditions.

Table 10. Reward-weight sensitivity of ATAC (RQ6), averaged across the three disaster-intensity levels and independent simulation seeds. Arrows indicate the preferred direction of each metric.

Reward Profile	TCR $\uparrow$	WTS $\uparrow$	LCC $\uparrow$	FR $\downarrow$	Energy Used $\downarrow$
Balanced	0.830	0.830	0.724	0.055	5.292
Task-oriented	0.879	0.864	0.818	0.040	4.466
Connectivity-oriented	0.781	0.779	0.810	0.042	4.654
Resilience-oriented	0.776	0.786	0.648	0.063	6.135
Energy-aware	0.851	0.858	0.790	0.043	4.060

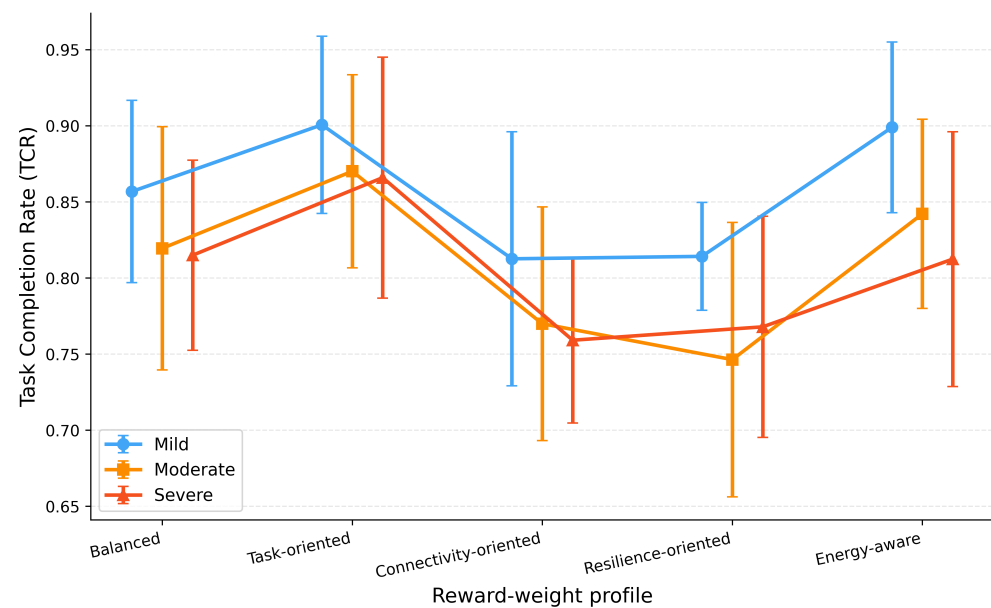


Figure 23. Sensitivity of ATAC task completion performance to the mission-level reward weights under mild, moderate, and severe disaster conditions. Points report the mean Task Completion Rate (TCR) across independent simulation seeds, and error bars indicate one standard deviation.

Overall, the reward-weight sensitivity analysis demonstrated that ATAC was not biased toward a particular reward configuration but had multi-objective trade-offs between the considered mission metrics. In particular, the task-oriented configuration was the most beneficial in terms of mission performance, while the energy-aware configuration was the most beneficial in terms of energy reduction. These results also show that increasing a particular reward component does not necessarily improve the corresponding system metric due to the complex interactions between different aspects of the UAV motion.

7.10. Overall Discussion

Taken together, the six simulation studies provide a more nuanced picture of ATAC than a claim of uniform superiority across all metrics. The main result is that adaptive k -truss coordination changes the operating point between mission execution, communication structure, and resource expenditure as the swarm and its communication topology evolve.

RQ1 and RQ2 provide the main evidence under communication and UAV degradation. Under link failures, adaptive ATAC maintains a high aggregate TCR (0.871 ± 0.058), outperforming GTA, CPC, and GAC, while remaining close to ATAC-Fixed (0.879 ± 0.052). A similar pattern is observed under UAV node failures, where adaptive ATAC achieves 0.857 ± 0.059 TCR compared with 0.868 ± 0.058 for ATAC-Fixed. More importantly, in both studies the selected truss level changes as the topology deteriorates. Lower values of k are used more frequently under stronger degradation, showing that the framework does not impose the same structural requirement throughout changing network conditions.

RQ3 clarifies why this adaptation is relevant. No fixed value of k is uniformly preferable across the evaluated metrics. Under the tested stationary condition, $k = 4$ provides the highest TCR and lowest fragmentation, whereas $k = 5$ provides the highest LCC and average triangle support. Adaptive ATAC is therefore not claimed to outperform the best retrospectively selected fixed k in every static scenario. Rather, the fixed- k sensitivity study shows that k controls a real mission–structure trade-off, while RQ1 and RQ2 show that ATAC can modify this operating point when conditions change.

The ablation results in RQ4 further support this interpretation. Relative to ATAC-Fixed, adaptive ATAC improves LCC, fragmentation, and ATS, while incurring a small reduction in aggregate TCR. At the same time, GTA and GAC remain preferable on some individual graph or runtime measures. The contribution of ATAC should therefore be understood as the joint use of graph-aware coordination, truss-based structural constraints, and adaptive structural selection, rather than as universal dominance on every metric.

RQ5 bounds the scalability claim to the range that was actually tested. ATAC was evaluated with swarms from 20 to 90 UAVs. Runtime increases with swarm size, but ATAC remains computationally executable throughout this range and has lower measured runtime per coordination step than GAC at each tested swarm size. At the largest configuration, ATAC also maintains strong connectivity and low fragmentation. These results support scalability within the evaluated range; they do not establish performance for substantially larger swarms or different hardware and network conditions.

Finally, RQ6 shows that the reported behavior is not tied to a single reward-weight vector. Changing the task, connectivity, structural, and energy weights produces distinct operating points across mission performance, topology, and resource consumption. In particular, the task-oriented profile provides the highest aggregate task completion performance, whereas the energy-aware profile gives the lowest modeled energy expenditure. The results also show that increasing one reward weight does not necessarily maximize the corresponding system-level metric, reflecting the coupling between mobility, task execution, and communication topology.

Overall, the simulation evidence supports ATAC as an adaptive coordination framework for situations in which mission execution and communication resilience must be considered jointly. Its main advantage is not that it maximizes every metric, but that it can modify the structural requirement as network conditions change while maintaining competitive mission performance and comparatively low modeled resource expenditure. This property is particularly relevant to disaster-response missions, where communication quality and UAV availability may vary during deployment. The conclusions remain limited to the simulated conditions considered here; correlated wireless failures, heterogeneous UAV capabilities, hardware effects, and real-flight validation remain important directions for future work.

7.11. Limitations and Future Research Directions

Although the simulation results provide evidence for the usefulness of adaptive k -truss coordination, several limitations define the scope within which the present conclusions should be interpreted.

First, the evaluation is entirely simulation-based. The communication, mobility, energy, and failure models provide controlled conditions for comparing the coordination strategies, but they do not capture all of the effects encountered by physical UAV platforms. In particular, the reported energy values are simulator-specific and should not be interpreted directly as physical battery endurance or flight time. Hardware-in-the-loop (HIL) evaluation and experiments with real UAVs are therefore necessary before drawing conclusions about field deployment.

Second, communication-link failures are modeled as independent stochastic events. This abstraction is useful for controlling the severity of communication degradation, but real disaster environments may produce spatially and temporally correlated failures due to obstruction, multipath fading, interference, congestion, or common infrastructure loss. The present study also does not consider adversarial communication attacks such as jamming or intentional link disruption. Extending the evaluation to correlated wireless failures and adversarial conditions is therefore an important direction for future work.

Third, the current simulations assume a homogeneous UAV swarm. Agents share the same basic mobility, communication, and resource model, whereas operational disaster-response teams may contain heterogeneous platforms with different sensing capabilities, communication ranges, battery capacities, payloads, and mobility constraints. Future work should investigate whether the adaptive structural policy remains effective when the available agents have substantially different capabilities.

Fourth, the current implementation reconstructs the selected k -truss backbone from the updated communication graph rather than maintaining it incrementally after individual edge insertions and deletions. This design simplifies the closed-loop implementation and ensures that the backbone is computed from the current topology, but its computational cost may become restrictive for substantially larger or more rapidly changing networks. Incremental or fully dynamic k -truss maintenance is therefore a natural extension of the present framework.

Fifth, the reward coefficients are fixed during each simulation run. RQ6 shows that changing these coefficients produces different operating points across task completion, connectivity, structural behavior, and modeled energy consumption; however, the current framework does not adapt the reward priorities online. The online adaptation investigated in this work concerns the graph state and the selected truss level k , not the reward coefficients themselves. Future work could examine mission-aware or context-dependent reward adaptation while preserving the structural constraints of the framework.

Finally, the scalability conclusions are limited to the swarm sizes actually evaluated in RQ5, namely 20–90 UAVs. Although ATAC remains computationally executable throughout this range, these results do not establish scalability to substantially larger swarms, denser communication graphs, or resource-constrained onboard processors. Future validation should therefore combine larger-scale simulations with hardware-in-the-loop tests and real multi-UAV experiments to evaluate computational latency, communication overhead, energy consumption, and structural adaptation under realistic disaster-response conditions.

These limitations indicate that the present results should be viewed as simulation-based evidence for the proposed coordination mechanism rather than as a complete validation of operational deployment. Future work will focus particularly on incremental backbone maintenance, correlated and adversarial communication failures, heterogeneous UAV teams, context-dependent reward design, and HIL and real-flight validation.

8. Conclusions

This paper presented ATAC, an Adaptive k -Truss-Constrained Agentic Coordination framework for resilient multi-agent UAV swarm coordination in dynamic disaster environments. The framework models the swarm as a dynamic communication–task graph and integrates adaptive k -truss backbone selection with graph-aware decentralized coordination. Rather than treating communication topology as a passive constraint, ATAC allows structural information from the evolving graph to influence local UAV decisions.

The simulation results show that the main benefit of ATAC lies in the balance between mission execution and communication structure, rather than in universal superiority on every individual metric. Under communication-link failures (RQ1), adaptive ATAC

achieves an aggregate TCR of 0.871 ± 0.058 , outperforming GTA, CPC, and GAC while remaining close to ATAC-Fixed. Under UAV node failures (RQ2), it similarly maintains a high aggregate TCR of 0.857 ± 0.059 . In both cases, the selected truss level changes as degradation becomes more severe, with lower values of k used more frequently when the communication topology can no longer support the stronger structural requirement.

The fixed- k sensitivity analysis (RQ3) further shows that no single truss level is uniformly preferable. Different values of k favor different mission and structural metrics, which supports treating k as a structural control variable rather than as a universally fixed parameter. The ablation study (RQ4) provides complementary evidence: relative to ATAC-Fixed, adaptive ATAC improves LCC, fragmentation, and average triangle support, with only a small reduction in aggregate TCR. These results characterize adaptation as a change in the mission–structure operating point rather than as a mechanism that maximizes every metric simultaneously.

The scalability study (RQ5) shows that ATAC remains computationally executable over the evaluated range of 20–90 UAVs. Runtime increases with swarm size, as expected, but remains below that of the graph-aware GAC baseline at every tested swarm size. Finally, the reward-sensitivity study (RQ6) shows that the observed behavior is not restricted to a single reward-weight vector. Different reward profiles produce distinct operating points across task completion, connectivity, structural behavior, and modeled energy expenditure.

Taken together, these results support ATAC as a graph-aware decentralized coordination framework for disaster-response settings in which mission progress, communication structure, and resource use must be considered jointly. The contribution is therefore not a claim that adaptive ATAC dominates every baseline or fixed- k configuration under all conditions. Rather, ATAC provides a mechanism for adjusting the structural requirement of the swarm as the communication topology changes, while maintaining competitive mission performance within the tested conditions.

The present evidence remains simulation-based and is limited to the failure models, swarm sizes, and system assumptions considered in this study. As discussed in Section 7.11, future work will extend the framework toward correlated and adversarial communication failures, heterogeneous UAV teams, incremental k -truss maintenance, larger-scale deployments, and hardware-in-the-loop and real-UAV validation.

Author Contributions: Conceptualization, H.H. and N.A.; Methodology, H.H. and N.A.; Software, H.H. and N.A.; Validation, H.H.; Formal analysis, H.H.; Investigation, H.H.; Resources, N.A.; Data curation, N.A.; Writing—original draft, H.H. and N.A.; Writing—review & editing, H.H. and N.A.; Visualization, H.H.; Supervision, H.H.; Project administration, H.H.; Funding acquisition, H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Deanship of Graduate Studies and Scientific Research at Jouf University under grant No. (DGSSR-2024-02-02073).

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request.

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

References

1. Erdelj, M.; Natalizio, E. UAV-assisted disaster management: Applications and open issues. *Comput. Netw.* **2016**, *93*, 459–476. [CrossRef]
2. Hayat, S.; Yanmaz, E.; Muzaffar, R. Survey on unmanned aerial vehicle networks for civil applications. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2624–2661. [CrossRef]
3. Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; Zhang, M.; Wang, J.; Jin, S.; Zhou, E.; et al. The rise and potential of large language model based agents: A survey. *Sci. China Inf. Sci.* **2025**, *68*, 121101. [CrossRef]

4. Ferrag, M.A.; Tihanyi, N.; Debbah, M. From LLM Reasoning to Autonomous AI Agents: A Comprehensive Review. *IEEE Access* **2026**, *14*, 84237–84285. [\[CrossRef\]](#)
5. Khan, A.; Gupta, S.; Gupta, S.K. Emerging UAV technology for disaster detection, mitigation, response, and preparedness. *J. Field Robot.* **2022**, *39*, 905–955. [\[CrossRef\]](#)
6. Wang, K.; Cheng, Z. Multi-UAV cooperative task scheduling and trajectory optimization system under communication constraints. *Phys. Commun.* **2026**, *76*, 103073. [\[CrossRef\]](#)
7. 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. [\[CrossRef\]](#)
8. Orr, J.; Dutta, A. Multi-agent deep reinforcement learning for multi-robot applications: A survey. *Sensors* **2023**, *23*, 3625. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Ekechi, C.C.; Elfouly, T.; Alouani, A.; Khattab, T. A survey on UAV control with multi-agent reinforcement learning. *Drones* **2025**, *9*, 484. [\[CrossRef\]](#)
10. Alam, M.M.; Trina, S.A.; Hossain, T.; Mahmood, S.; Ahmed, M.S.; Arafat, M.Y. Variations in Multi-Agent Actor–Critic Frameworks for Joint Optimizations in UAV Swarm Networks: Recent Evolution, Challenges, and Directions. *Drones* **2025**, *9*, 153. [\[CrossRef\]](#)
11. Wang, Z.; Yao, H.; Mai, T.; Xiong, Z.; Wu, X.; Wang, S. Learning-based routing in UAV swarm networks: A multi-agent reinforcement learning approach. *IEEE Trans. Veh. Technol.* **2022**, *71*, 11050–11062. [\[CrossRef\]](#)
12. Ding, Y.; Yang, Z.; Pham, Q.V.; Hu, Y. Distributed Machine Learning for UAV Swarms: Computing, Sensing, and Semantics. *IEEE Internet Things J.* **2023**, *11*, 7447–7473. [\[CrossRef\]](#)
13. Wang, J.; Cheng, J. Truss decomposition in massive networks. *Proc. VLDB Endow.* **2012**, *5*, 812–823. [\[CrossRef\]](#)
14. Conte, A.; De Sensi, D.; Grossi, R.; Marino, A. Truly scalable k-truss and max-truss algorithms for community detection in graphs. *IEEE Access* **2020**, *8*, 139096–139109. [\[CrossRef\]](#)
15. Rezvani, M.; Rezvani, M. Truss decomposition using triangle graphs. *Soft Comput.* **2022**, *26*, 55–68. [\[CrossRef\]](#)
16. Zheng, Z.; Ye, F.; Li, R.H.; Ling, G.; Jin, T. Finding weighted k-truss communities in large networks. *Inf. Sci.* **2017**, *417*, 344–360. [\[CrossRef\]](#)
17. Zhu, W.; Zhang, M.; Chen, C.; Wang, X.; Zhang, F.; Lin, X. Pivotal relationship identification: The k-truss minimization problem. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI), Macao, China, 10–16 August 2019; pp. 4874–4880. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Luo, Q.; Yu, D.; Cheng, X.; Cai, Z.; Yu, J.; Lv, W. Batch Processing for Truss Maintenance in Large Dynamic Graphs. *IEEE Trans. Comput. Soc. Syst.* **2020**, *7*, 1435–1446. [\[CrossRef\]](#)
19. Luo, Q.; Yu, D.; Cheng, X.; Sheng, H.; Lv, W. Exploring Truss Maintenance in Fully Dynamic Graphs: A Mixed Structure-Based Approach. *IEEE Trans. Comput.* **2023**, *72*, 707–718. [\[CrossRef\]](#)
20. Sun, Z.; Huang, X.; Xu, J.; Bonchi, F.; Chang, L. Probabilistic truss decomposition on uncertain graphs: Indexing and Dynamic Maintenance. *ACM Trans. Knowl. Discov. Data* **2025**, *50*, 6. [\[CrossRef\]](#)
21. Ning, Z.; Yang, Y.; Wang, X.; Song, Q.; Guo, L.; Jamalipour, A. Multi-Agent Deep Reinforcement Learning Based UAV Trajectory Optimization for Differentiated Services. *IEEE Trans. Mob. Comput.* **2024**, *23*, 5818–5834. [\[CrossRef\]](#)
22. 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 Proceedings of the ICC 2025—IEEE International Conference on Communications; IEEE: Piscataway, NJ, USA, 2025, pp. 1–6. [\[CrossRef\]](#)
23. López-Villegas, I.; Medina-Gómez, K.J.; Izquierdo-Reyes, J.; Colin-García, D.; González-Hernández, H.G.; Bustamante-Bello, R. A Transformer-Based Self-Organizing UAV Swarm for Assisting an Emergency Communications System. *Drones* **2025**, *9*, 769. [\[CrossRef\]](#)
24. Peng, Y.; Zhou, C.; Cui, H.; Duan, T.; Chen, H.; Zhang, F.; Liu, S. *Resilient UAV Swarm with Fast Connectivity Recovery and Extensive Coverage*; AAAI Press: Washington, DC, USA, 2026; Volume 40, pp. 917–925. [\[CrossRef\]](#)
25. Liu, J.; Yang, C.; Lu, Z.; Chen, J.; Li, Y.; Zhang, M.; Bai, T.; Fang, Y.; Sun, L.; Yu, P.S.; et al. Graph Foundation Models: Concepts, Opportunities and Challenges. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 5023–5044. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Shi, C.; Chen, J.; Liu, J.; Yang, C. Graph foundation model. *Front. Comput. Sci.* **2024**, *18*, 186355. [\[CrossRef\]](#)
27. Zeng, L.; Ye, S.; Chen, X.; Zhang, X.; Ren, J.; Tang, J.; Yang, Y.; Shen, X. Edge Graph Intelligence: Reciprocally Empowering Edge Networks With Graph Intelligence. *IEEE Commun. Surv. Tutor.* **2025**, *27*, 3417–3454. [\[CrossRef\]](#)
28. Aguzzi, G.; Domini, D.; Venturini, F.; Viroli, M. Scaling Swarm Coordination with GNNs—How Far Can We Go? *AI* **2025**, *6*, 282. [\[CrossRef\]](#)
29. Poghosyan, S.; Poghosyan, V.; Abrahamyan, S.; Lazyan, A.; Astsatryan, H.; Alaverdyan, Y.; Eguiazarian, K. Cloud-based mathematical models for self-organizing swarms of UAVs: Design and analysis. *Drone Syst. Appl.* **2024**, *12*, 1–12. [\[CrossRef\]](#)

30. Xiao, J.; Fang, X.; Jia, Q.; Feroskhan, M. Learning Resilient Formation Control of Drones with Graph Attention Network. *IEEE Internet Things J.* **2025**, *12*, 24028–24040. [[CrossRef](#)]
31. Pedroso, C.; Santos, A. A Survey on Faults and Vulnerabilities in Network Topological Connectivity: Logical and Physical Perspectives. *ACM Comput. Surv.* **2025**, *58*, 125. [[CrossRef](#)]

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.