

## Article

# CogMUS: A Soar-Based Cognitive Framework for Mission Understanding in Multi-UAV Cooperative Operation

Jiaxin Hu <sup>1</sup>, Tao Wang <sup>1,2,\*</sup>, Hongrun Wang <sup>1</sup> and Jingshuai Cao <sup>1</sup>

<sup>1</sup> School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen 518107, China; hujx57@mail2.sysu.edu.cn (J.H.); wanghr56@mail2.sysu.edu.cn (H.W.); caojsh3@mail2.sysu.edu.cn (J.C.)  
<sup>2</sup> Southern Marine Science and Engineering Guangdong Laboratory, Zhuhai 519000, China  
\* Correspondence: wangt339@mail.sysu.edu.cn

## Highlights

### What are the main findings?

- This study presents CogMUS, a cognitive framework for multi-UAV cooperative mission understanding that leverages the working memory, long-term memory and decision cycle of the Soar cognitive architecture to enable critical autonomous capabilities.
- The CogMUS demonstrates high adaptability and robustness across diverse and dynamic mission scenarios, effectively translating abstract high-level commands into coordinated, executable actions for the UAV swarm.

### What are the implications of the main findings?

- Experimental results demonstrate that across five complex collaborative scenarios, CogMUS achieves an average task understanding accuracy and task completion rate exceeding 80%, with average cooperative efficiency exceeding 90%.
- This study highlights the potential of cognitive architectures to effectively bridge the gap between high-level human intent and autonomous multi-agent execution, and promote the development of more transparent, reliable and intelligent UAV swarm.

## Abstract

The cooperative operation of multiple Unmanned Aerial Vehicles (multi-UAV) is emerging as a pivotal trend in future complex autonomous systems. To enable accurate mission understanding and efficient collaboration among UAVs in complex, dynamic, and uncertain operational environments, this paper introduces CogMUS, a novel cooperative mission understanding framework based on the Soar cognitive architecture. We first construct a mission understanding framework for UAV operations centered around five typical mission categories. Building on this foundation, we design a distributed cognitive model where each UAV is equipped with a Soar agent. This model leverages the synergy of working memory (WM), long-term memory (LTM), and the decision cycle (DC) to achieve key functionalities, including hierarchical mission decomposition, dynamic task allocation, and proactive airspace conflict detection and resolution. Through comprehensive simulation experiments, we validate the performance of the proposed CogMUS framework across key metrics, including task understanding accuracy, cooperative efficiency, and overall task completion rate. The results demonstrate that CogMUS exhibits superior adaptability to diverse scenarios, as well as remarkable scalability and robustness.

**Keywords:** multi-UAV; mission understanding; cognitive architecture; task decomposition; cooperative operation; autonomous systems



Academic Editor: Pablo Rodríguez-González

Received: 11 October 2025

Revised: 14 November 2025

Accepted: 20 November 2025

Published: 24 November 2025

**Citation:** Hu, J.; Wang, T.; Wang, H.; Cao, J. CogMUS: A Soar-Based Cognitive Framework for Mission Understanding in Multi-UAV Cooperative Operation. *Drones* **2025**, *9*, 813. <https://doi.org/10.3390/drones9120813>

**Copyright:** © 2025 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) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the landscape of unmanned autonomous systems, multi-UAV systems have emerged as a pivotal force for executing complex cooperative missions, owing to their inherent advantages such as distributed nature, high robustness, and cost-effectiveness [1,2]. The rapid advancement of artificial intelligence, particularly the deep integration of cognitive science with unmanned systems, has shifted the research focus toward enhancing the autonomous collaborative capabilities of UAV swarm in dynamic and uncertain environments [3–5]. Among these capabilities, the rapid and accurate understanding of complex cooperative missions is a fundamental prerequisite for intelligent decision-making and action, representing a core challenge in current research [6,7]. Effective mission understanding directly determines whether a multi-UAV system can accurately perceive the operational situation, grasp the high-level intent, and subsequently execute efficient task allocation and cooperative operation [8,9].

However, achieving high-level mission understanding in multi-UAV systems faces several domain-specific challenges: (1) Deep Semantic Comprehension of High-Level Intent: High-level mission directives are often abstract and goal-oriented (e.g., “monitor and secure Area A from unauthorized intrusions”). Translating such high-level intent into a series of concrete, executable, and coordinated actions for multiple UAVs is a significant obstacle, especially under the constraints of onboard computational resources [10,11]. (2) Rapid and Accurate Comprehension of Dynamic and Incomplete Information: The operational environment is inherently volatile. UAVs must process heterogeneous data streams from diverse sensors, contend with communication latencies and packet loss, and construct a coherent situational awareness from partial and sometimes conflicting information to continuously update their understanding of the mission context [12,13]. (3) Efficient Multi-Agent Coordination and Deconfliction: Effective collaboration requires not only individual task execution but also a shared understanding of roles, responsibilities, and interdependencies. This involves complex negotiation, dynamic resource allocation, and proactive deconfliction of actions in three-dimensional space to prevent collisions and resource wastage [14,15].

To address these challenges, this paper focuses on the core problem of mission understanding in multi-UAV cooperative operation and proposes CogMUS, a novel cognitive mission understanding framework based on the Soar cognitive architecture [16]. Drawing inspiration from human cognitive decision-making mechanisms, CogMUS leverages the strengths of the Soar architecture in symbolic representation, problem-solving, and structured learning. By integrating knowledge engineering methods, we aim to construct a cognitive framework capable of rapid and accurate situational awareness and assessment, as well as deep comprehension and decomposition of cooperative missions. Through this approach, we expect to enhance the mission understanding capabilities and collaborative operational effectiveness of multi-UAV systems in complex and dynamic environments.

The main contributions of this paper are threefold:

- We propose a hierarchical multi-UAV cooperative mission understanding framework, structured around five typical operational mission archetypes, which provides a structured foundation for knowledge representation and reasoning.
- We design and implement CogMUS, a distributed cognitive model where each UAV agent utilizes the Soar architecture to interpret mission intent, perform hierarchical mission decomposition, and engage in sophisticated coordination to align actions and resolve conflicts.
- We conduct extensive simulation experiments, including performance analysis and ablation studies, demonstrating that our proposed method significantly outperforms mainstream baselines in task understanding accuracy (TUA), cooperative efficiency

(CE), and task completion rate (TCR), particularly in dynamic and highly complex environments, thereby validating the effectiveness of applying cognitive architectures to this problem domain.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive review of related work. Section 3 presents the formal problem definition. Section 4 details the proposed CogMUS framework and its core components. Section 5 presents the experimental setup, results, and analysis. Section 6 discusses the key findings and limitations of the study. Finally, Section 7 concludes the paper and outlines future research directions.

## 2. Related Work

### 2.1. Multi-UAV Cooperative Task Allocation and Control

Coordination technologies for multi-UAV systems have been extensively studied. Classical approaches often formulate task assignment as an optimization problem, such as the multiple traveling salesman problem (mTSP) [17] or the vehicle routing problem (VRP) [18], and employ techniques like mixed-integer linear programming (MILP) for solutions [19,20]. Market-based or auction-based mechanisms offer a decentralized alternative, where UAVs bid for tasks based on their capabilities and local utility functions, facilitating distributed decision-making [21]. Bio-inspired algorithms, including ant colony optimization (ACO) and particle swarm optimization (PSO), have also been applied to find near-optimal solutions for complex allocation problems [22]. In recent years, multi-agent reinforcement learning (MARL) has gained increasing attention, enabling UAVs to learn collaborative policies through trial-and-error in simulated environments [23–26]. Despite its power, MARL often faces challenges in scalability, convergence in vast state-action spaces, and the “black-box” nature of learned policies, which makes verification and trust difficult. Furthermore, most of these methods assume that tasks are pre-defined and well-structured, overlooking the critical preceding step of understanding and decomposing high-level mission intent.

Furthermore, a rich body of work exists in organizational-centered multi-agent systems, with frameworks like MOISE [27], which focuses on specifying, analyzing, and executing agent organizations, and platforms like Jacamo [28], which integrates agent-oriented programming with organizational concepts. These systems provide robust structures for defining roles, norms, and interaction protocols. However, they often assume that the mission has already been decomposed into well-defined roles and tasks. Our work with CogMUS complements these approaches by focusing on the preceding cognitive challenge: how a team of agents can collectively interpret a high-level, abstract command and autonomously derive the very tasks, roles, and coordination strategies that frameworks like MOISE and Jacamo would then manage.

### 2.2. Mission Understanding in Autonomous Systems

The field of mission understanding, or intent recognition, aims to bridge the gap between human instruction and machine execution. A significant body of research has focused on natural language processing (NLP), enabling robots or UAVs to parse free-form text or speech commands [10,29–31]. These methods typically rely on semantic parsing to map linguistic structures to executable actions. Another line of research utilizes formal knowledge representations, such as ontologies and knowledge graphs, to model domain concepts, tasks, and their relationships, thereby enabling logical reasoning about mission objectives [32,33]. Inverse reinforcement learning and learning from demonstration attempt to infer the underlying reward functions or policies from expert demonstrations, thus capturing implicit high-level intent [34,35]. However, these approaches are often designed

for single agents or simple human–robot interactions and do not address the complexities of distributed understanding, where multiple autonomous agents must establish and maintain a shared mission context [36].

Recently, Large Language Models (LLMs) have emerged as powerful tools for high-level task planning and human–robot interaction [37–39]. These models can interpret nuanced human intent far beyond the scope of traditional NLP methods. However, systems relying primarily on LLMs for agent control often operate in a more deliberative, open-loop manner and can face challenges with the real-time decision-making required in dynamic, uncertain environments [40,41]. Our work, based on the Soar cognitive architecture, emphasizes a tight perception-action loop via its rapid decision cycle, offering a complementary approach focused on continuous, closed-loop reasoning and reactivity.

### 2.3. Cognitive Architectures in Robotics and UAVs

Cognitive architectures provide unified theories of cognition for building human-like intelligent agents. Soar [16] and ACT-R [42] are two of the most prominent examples. They have been successfully applied to various single-agent tasks in robotics, such as navigation, manipulation, and human–robot interaction, demonstrating their capabilities in complex reasoning, planning, and learning [43–45]. In the UAV domain, cognitive architectures have been used to model pilot decision-making processes and to enable higher degrees of autonomous control for individual UAVs [46–48]. However, the application of these architectures to multi-agent systems, particularly for complex cooperative mission scenarios involving multiple UAVs, remains in its nascent stages. Existing multi-agent applications are often limited to simpler collaborative tasks [49,50].

Our work is distinguished from existing literature by developing a fully Soar-based distributed system that specifically addresses the complex problem of cooperative mission understanding in dynamic operational environments, with a strong emphasis on hierarchical decomposition and robust collaboration.

## 3. Problem Definition

In the context of cooperative operation, multi-UAV mission understanding is defined as the capability of a UAV swarm to accurately and deeply infer the intent, objectives, constraints, role divisions, coordination patterns, and expected outcomes of a given mission. This inference is based on both explicit and implicit information derived from high-level directives, situational awareness, and prior knowledge.

The core problem addressed in this paper is to develop a distributed cognitive process such that, given  $T$  and each UAV's local observation  $o_i(t) \subseteq E(t)$ , the system can:

1. Decompose the global mission  $T$  into a set of spatiotemporally coordinated subtasks,  $T = T_1, T_2, \dots, T_K$ .
2. Allocate these subtasks to individual UAVs or subgroups, forming an assignment scheme  $A : T \rightarrow \mathcal{P}(U)$  that maps each subtask to a subset of UAVs.
3. Generate a set of low-level, executable action plans for each UAV,  $P_i = \{a_{i,1}, a_{i,2}, \dots\}$ , where  $P_i$  is the action sequence for UAV  $u_i$ .

An effective policy must optimize multiple objectives, including maximizing TCR, minimizing completion time and resource consumption, and ensuring robustness against environmental uncertainties and objective actions. Mission understanding is a dynamic, iterative process that continuously updates and deepens as the operational situation evolves and new information becomes available.

## 4. Methodology

### 4.1. Overall Framework

To enable effective understanding of complex cooperative missions by multi-UAV systems, we have constructed a generalizable and extensible mission understanding framework, as depicted in Figure 1. This framework is based on the abstraction and analysis of cooperative missions that multi-UAV systems may execute in future complex scenarios. It serves as the foundational architecture for our proposed method, guiding the Soar agents in classifying, decomposing, and executing missions.

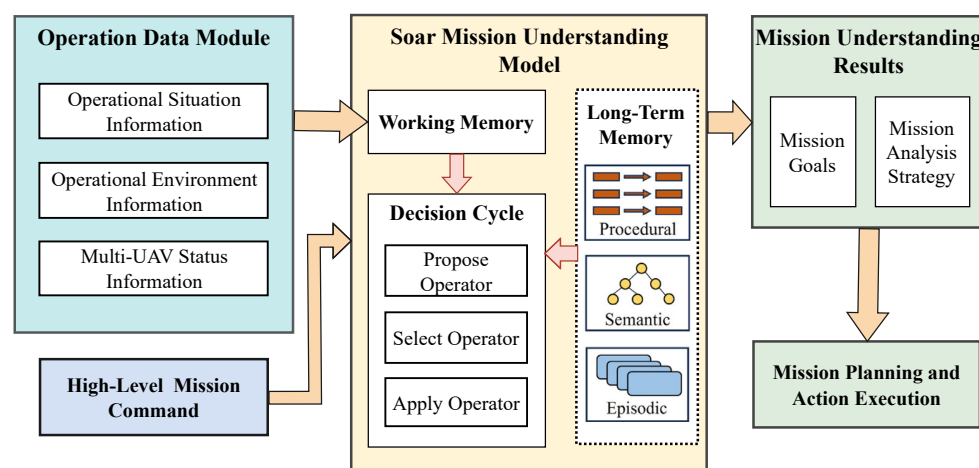


Figure 1. Overall framework of CogMUS.

To facilitate a structured understanding of complex cooperative missions, we define five typical operational mission archetypes, as detailed in Table 1. These archetypes span various operational phases, from monitoring and patrol to defense and support, comprehensively reflecting the complexity and diversity of cooperative operation.

Table 1. Typical cooperative missions and their understanding elements.

| Mission Category          | Main Objective(s)                                        | Key Information Requirements                       | Soar's Focus                                                                   |
|---------------------------|----------------------------------------------------------|----------------------------------------------------|--------------------------------------------------------------------------------|
| Cooperative Monitoring    | Obtain environmental information, monitor areas/targets  | Area, target features, potential hazards           | Infer information gaps, path optimization                                      |
| Cooperative Patrol        | Patrol the designated area, detect and report anomalies  | Patrol routes, area boundaries, anomaly signatures | Optimize patrol coverage, detect deviations, classify and respond to anomalies |
| Cooperative Defense       | Protect designated assets, deter unauthorized intrusions | Defensive area, anomaly levels                     | Prioritize anomalies, plan response trajectories                               |
| Cooperative Communication | Communication relay, network coverage                    | Network topology, data throughput requirements     | Evaluate network effectiveness, manage data routing                            |
| Cooperative Support       | Transport supplies, supplement energy                    | Support requirements, operational constraints      | Optimize resource allocation, secure support routes                            |

Based on the aforementioned typical missions, our mission understanding framework features a hierarchical and modular structure, shown in Figure 1. The Operation Data

Layer receives raw sensor data and environmental information, feeding it into Soar's WM. The intermediate Soar Mission Understanding Layer performs a deep analysis of the mission intent by analyzing situational elements and matching them against high-level directives, generating concrete action plans and coordination strategies. Each UAV's Soar agent reasons through its core decision cycle (Input-Propose-Select-Apply-Output). The final operator selected within the DC is outputted to the lower Mission Planning and Action Execution Layer. The UAV then executes this action, thereby altering the current state.

The central idea of this framework is that any specific mission can be mapped to one or a combination of these five archetypes, allowing for efficient mission understanding and response using pre-built knowledge and strategies. The extensibility of this framework lies in its ability to adapt to new mission types by adding or modifying the corresponding mission templates and Soar rule sets.

#### 4.2. Soar Cognitive Model

The Soar cognitive architecture is a general model of artificial intelligence designed to simulate human cognitive processes. It consists of three core sub-modules: WM, LTM, and DC. The workflow of these modules is illustrated in Figure 2. These three components work in concert to form the mission understanding capability. Our framework operates on a fully distributed control paradigm, each UAV agent runs its own Soar instance and makes decisions autonomously. The collective or "shared" understanding of the mission emerges from the continuous communication among these distributed agents. Each agent's local WM is updated not only by its own perceptions but also by processing messages from teammates. This ensures that coordination is a dynamic and emergent property of the swarm's interactions, guided by shared protocols encoded in each agent's LTM, rather than a pre-scripted or centrally commanded process. The state of each Soar agent can be represented as:

$$S_i(t) = \langle WM_i(t), LTM_i(t), DC_i(t) \rangle \quad (1)$$

where  $WM_i(t)$ ,  $LTM_i(t)$ , and  $DC_i(t)$  represent the states of the WM, LTM, and DC for the  $i$ -th agent at time  $t$ , respectively.

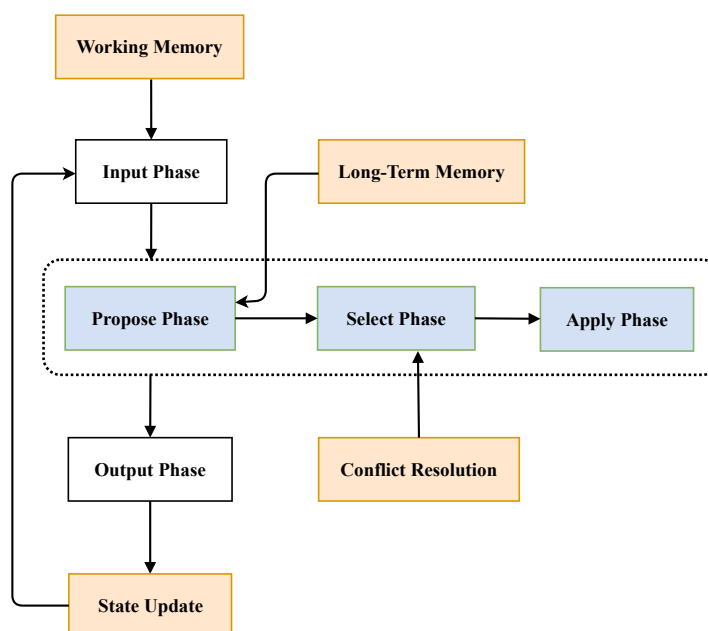


**Figure 2.** Internal workflow of the Soar sub-modules. The arrows depict the sequential flow of the decision cycle, and the dashed box encloses the core cognitive components.

The collective cognitive capability of the system emerges from the communication and coordination among these agents:

$$\mathcal{C}(t) = \left( \bigcup_{i=1}^n S_i(t) \right) \oplus \mathcal{J}(t) \quad (2)$$

where  $\mathcal{J}(t)$  represents the joint information space at time  $t$ , which contains emergent knowledge and shared context that arises from the communication and coordination among agents. The operator  $\oplus$  denotes a synergistic integration, representing the process by which the aggregated individual cognitive states are combined with the joint information space to produce an emergent collective capability that is greater than the sum of its parts.

#### 4.2.1. Working Memory Module

WM is the agent's short-term dynamic memory for storing the current understanding of the situation. It dynamically stores and processes real-time operational information in the form of symbolic knowledge units called Working Memory Elements (WMEs). Each WME is represented as a triple:  $\langle \text{id} \rangle \wedge \langle \text{attribute} \rangle \langle \text{value} \rangle$ .

The dynamic update process of WM can be modeled as:

$$WM(t+1) = \alpha \cdot WM(t) + \beta \cdot P(t) + \gamma \cdot I(t) \quad (3)$$

where  $P(t)$  represents the set of new WMEs derived from external perceptions and communication at time  $t$ , and  $I(t)$  represents the set of new WMEs generated internally through the agent's reasoning and decision-making process.  $\alpha$  is the historical information retention coefficient,  $\beta$  is the new information fusion coefficient,  $\gamma$  is the inferred information weight, subject to the constraint  $\alpha + \beta + \gamma = 1$ .

For example, a high-level directive for a cooperative monitoring mission over a designated area can be represented in WM as follows:

```
(S1 ^state-link <S1>)
(<S1> ^mission <M1>)
(<M1> ^type cooperative-monitoring
  ^area <A1>
  ^priority medium
  ^duration 120
  ^monitoring-type persistent-surveillance)
(<A1> ^type geographical-zone
  ^boundary-points <P1> <P2> <P3> <P4>
  ^known-anomalies <T1>)
(<P1> ^lat 35.201 ^lon -117.850)
(<P2> ^lat 35.201 ^lon -117.800)
(<P3> ^lat 35.180 ^lon -117.800)
(<P4> ^lat 35.180 ^lon -117.850)
(<T1> ^type mobile-sam-launcher)
```

#### 4.2.2. Long-Term Memory Module

LTM stores the domain knowledge, coordination strategies, and learned experiential rules required for mission execution. In Soar, LTM is primarily organized in the form of production rules, which support efficient pattern matching and knowledge retrieval. LTM stores the agent's procedural knowledge as If-Then production rules. This knowledge base encodes everything from high-level instructions to decision heuristic rules and coordination strategies. Below are some rule examples:

- Rule Example 1: Mission Decomposition

```
sp {propose*decompose-patrol-mission-on-anomaly
```

```

(state <s> ^mission <m>)
(<m> ^type cooperative-patrol)
(<s> ^sensor-input <sensor>)
(<sensor> ^object-detected <obj>
^status unverified)
(<obj> ^is-anomaly true)
-->
(<s> ^operator <o> +)
(<o> ^name handle-patrol-anomaly
^subtask-1 <st1>
^subtask-2 <st2>)
(<st1> ^type investigate-anomaly
^target <obj>)
(<st2> ^type notify-teammates
^info <obj>
^urgency high)
}

```

This rule states that if the current mission is a cooperative patrol and an unverified anomaly is detected by UAV swarm, it proposes an operator to decompose it into two parallel subtasks: a primary subtask for the detecting UAV to investigate the anomaly (e.g., approach for a closer look and classify it), and a secondary subtask to notify all teammates in the swarm about the anomaly's detection and location.

- Rule Example 2: Airspace Deconfliction

```

sp {propose*deconflict-flight-path
(state <s> ^self <u1> ^teammate <u2>)
(<u1> ^flight-path <p1>)
(<u2> ^flight-path <p2>)
(compute-intersection <p1> <p2> |>| 0)
(<u1> ^priority <pr1>)
(<u2> ^priority <pr2>)
(<pr1> < <pr2>)
-->
(<s> ^operator <o> +)
(<o> ^name alter-flight-path
^reason deconfliction
^new-altitude (+ <u1>.altitude 500ft))
}

```

This rule identifies a potential flight path intersection with a higher-priority teammate and proposes an operator to alter its altitude to deconflict. In this rule, compute-intersection <p1> <p2> is a function that takes two flight paths and returns a boolean value indicating whether they will intersect in spacetime within a predefined safety margin. The priority attribute is an integer value assigned to each UAV, typically ranging from 1 (lowest) to 10 (highest), determined by its current role in the mission. The altitude change (+ <u1>.altitude 500) represents a command to increase the current altitude by a fixed value.

#### 4.2.3. Decision Cycle Module

The DC is Soar's execution engine, operating at a high frequency through a continuous "Input-Propose-Select-Apply-Output" loop, enabling the agent to respond to changes in

real-time. This cycle matches LTM rules against the current WM content, proposes operators (actions), selects the best operator based on preferences, applies its changes to WM, and sends commands to the execution layer, thus closing the loop from perception to action.

- **Phase 1: Input.** The input phase is responsible for receiving external perceptual information and internal state updates and integrating them into WM. The input processing function is defined as:

$$Input(t) = \phi(Perc(t), Int(t), Comm(t)) \quad (4)$$

where  $\phi$  is the information fusion function,  $Perc(t)$  is perceptual information,  $Int(t)$  is the internal state, and  $Comm(t)$  is the set of incoming communication messages from teammates.

- **Phase 2: Propose.** In this phase, the Soar engine matches the current content of WM against the conditions of all production rules in LTM. All rules whose conditions are met are activated in parallel. Each activated rule can propose one or more operators (i.e., potential actions or subgoals) as candidates for execution. This results in a set of candidate operators  $O_{candidate}$ , representing all plausible next steps given the current situation. The set of candidate operations is then:

$$O_{candidate} = \cup_{r_i \in R_{active}} \cdot Action(r_i) \quad (5)$$

- **Phase 3: Select.** If the Propose phase generates more than one candidate operator, the Select phase resolves the choice. This is achieved using preferences (e.g., ‘better’, ‘worse’), which are also generated by rules in LTM. The decision mechanism evaluates these preferences to select a single ‘best’ operator  $o^*$  from the candidate set. If an impasse occurs (i.e., no clear choice can be made), Soar creates a substate to reason about the selection. The selection is guided by an evaluation function:

$$o^* = \arg \max_{o \in O_{candidate}} \sum_{k=1}^K w_k \cdot f_k(o) \quad (6)$$

where  $f_k(o)$  is the  $k$ -th evaluation function, and  $w_k$  is its corresponding weight.

- **Phase 4: Apply.** The apply phase executes the selected operation and updates the system state:

$$State(t+1) = Apply(State(t), o^*) \quad (7)$$

- **Phase 5: Output.** The output phase transforms the decision results into specific execution commands, which are sent to the execution module.

#### 4.3. Cooperative Mission Decomposition Algorithm

Within the Soar cognitive model constructed above, we employ a cooperative mission decomposition algorithm to break down complex high-level missions. Let the original mission be  $T_0$  and the number of decomposition levels be  $L$ . The mission decomposition process can be represented as:

$$T_0 \xrightarrow{\text{decompose}} T_1^1, T_1^2, \dots, T_1^{n_1} \xrightarrow{\text{decompose}} \dots \xrightarrow{\text{decompose}} T_L^1, T_L^2, \dots, T_L^{n_L} \quad (8)$$

The decomposition function is defined as:

$$Decompose(T_i^j) = T_{i+1}^k | k \in SubTasks(T_i^j) \quad (9)$$

where  $SubTasks(T_i^j)$  is the set of submissions for mission  $T_i^j$ .

The task decomposition process is not an independent and static algorithm, but rather emerges from the continuous operation of the Soar's DC, guided by the decomposition rules in LTM. When a high-level mission is placed in WM, decomposition rules are triggered, proposing operators that create subgoals and subtasks. This process continues recursively until the complex mission is decomposed into atomic tasks executable by a single UAV, as described in Algorithm 1.

---

**Algorithm 1:** Soar-Based Mission Decomposition Algorithm

---

**Input:** High-level mission  $T$ , UAV set  $U$ , Current situation  $S_t$   
**Output:** Task assignment scheme  $A$

```

1 WM  $\leftarrow \{T, U, S_t\}$ , TaskQueue  $Q \leftarrow \{T\}$ ,  $A \leftarrow \emptyset$ 
2 while  $Q \neq \emptyset$  do
3    $task \leftarrow \text{DEQUEUE}(Q)$ 
4    $R_{task} \leftarrow \text{MATCH}(task, \text{ProductionRules})$ 
5   if  $|R_{task}| > 1$  then
6      $priorities \leftarrow \text{COMPUTE PRIORITY}(R_{task})$ 
7      $r^* \leftarrow \text{argmax}(priorities)$ 
8   else if  $|R_{task}| = 1$  then
9      $r^* \leftarrow R_{task}[0]$ 
10  else
11     $subgoal \leftarrow \text{GENERATE SUBGOAL}(task, \text{"find-decomposition"})$ 
12     $\text{ENQUEUE}(Q, subgoal)$ 
13    continue
14   $subtasks \leftarrow \text{APPLY}(r^*, task)$ 
15  foreach  $subtask$  in  $subtasks$  do
16    if  $\text{IS ATOMIC}(subtask)$  then
17       $agent \leftarrow \text{ASSIGN OPTIMAL}(subtask, U)$ 
18       $A \leftarrow A \cup \{(subtask, agent)\}$ 
19    else
20       $\text{ENQUEUE}(Q, subtask)$ 
21  $A_{\text{optimal}} \leftarrow \text{OPTIMIZE ASSIGNMENT}(A, \text{Constraints})$ 
22 return  $A_{\text{optimal}}$ 

```

---

## 5. Experiments and Results

### 5.1. Experimental Setup

We constructed a multi-UAV cooperative operation simulation environment using the AnyLogic platform, as illustrated in Figure 3. In AnyLogic, each UAV is modeled as an independent agent equipped with perception, Soar cognitive, and communication modules. The integration between the Soar cognitive architecture and the AnyLogic simulation platform was achieved through a custom-built Java-based middleware. For each UAV agent modeled in AnyLogic, a corresponding Soar kernel was instantiated and run in a separate thread. We utilized the Java Soar Interface to connect the two components. The AnyLogic model was responsible for simulating the physical world, including sensor inputs and the communication network. At each simulation step, the middleware collected perceptual data from the AnyLogic environment, translated it into symbolic WMEs, and fed it into the Soar agent's input-link. Conversely, commands generated by the Soar agent's decision cycle (i.e., selected operators) were retrieved from the output-link, translated into

executable actions (e.g., ‘set-velocity’, ‘change-altitude’), and passed to the corresponding UAV agent in AnyLogic for execution.

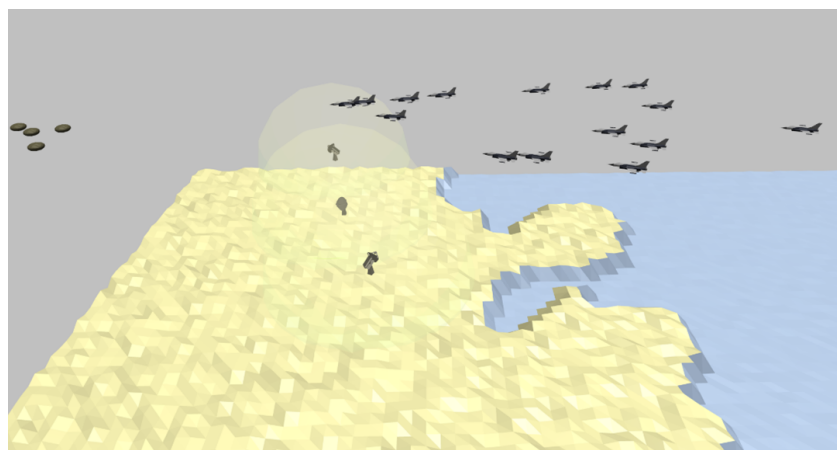


**Figure 3.** An example of the multi-UAV cooperative operational simulation environment built on the AnyLogic platform. This scenario depicts 15 UAVs executing a cooperative defense mission to protect four fixed facilities.

The experimental environment was configured with: (1) Mission Scenario: A dynamic operational area containing various targets and obstacles (e.g., stationary defensive systems). (2) Mission Types: A mix of missions including cooperative monitoring, patrol, and defense to test the method’s adaptability and effectiveness. (3) UAV Swarm: The number of UAVs ranged from 3 to 15, each equipped with a Soar agent capable of autonomous mission understanding and decision-making. Detailed experimental parameters are provided in Table 2. The simulations were conducted on a workstation equipped with an Intel Core i9-12900K CPU, 64 GB of RAM, and an NVIDIA GeForce RTX 3090 GPU. The AnyLogic simulation environment was version 8.7. The Soar agents were implemented using the official Soar 9.6 release and the Java Soar Interface.

**Table 2.** Experimental parameter settings.

| Parameter Category    | Parameter Name          | Specification                                    |
|-----------------------|-------------------------|--------------------------------------------------|
| Operation Environment | Simulation Area         | 50 km × 50 km square operational area            |
|                       | Target Distribution     | 5–8 dynamic/static targets, randomly distributed |
| UAV Swarm             | Number of UAVs          | 3–15                                             |
| Soar Agent            | WM Capacity             | 100 WMEs                                         |
|                       | LTM Rule Count          | 500–800 production rules                         |
|                       | DC Time                 | 50 ms                                            |
| Mission Complexity    | Max Subtasks            | 8                                                |
|                       | Max Decomposition Depth | 3 levels                                         |

## 5.2. Evaluation Metrics and Baselines

To comprehensively evaluate the performance of our proposed mission understanding method, we adopted three metrics: TUA, CE, and TCR, calculated as follows:

$$TUA = \frac{N_c}{N_t} \times 100\%, CE = \frac{T_{op}R_u}{T_{ac}R_t} \times 100\%, TCR = \frac{N_{co}}{N_{as}} \times 100\% \quad (10)$$

where  $N_c$  is the number of correctly understood missions,  $N_t$  is the total missions;  $T_{op}$  is the theoretical optimal completion time, which is established for each scenario by using an offline oracle solver on a simplified version of the problem with complete information

and no communication delays.  $T_{ac}$  is the actual completion time;  $N_{co}$  is the number of successfully completed missions, and  $N_{as}$  is the total assigned missions.

These metrics were selected not only to measure performance but also to serve as proxies for operational robustness and safety. A high TUA indicates a reduced risk of misunderstanding commands, while a high CE suggests smoother operations with fewer conflicts and wasted resources, directly contributing to a lower risk profile for the mission.

We selected two mainstream methods as baselines for comparison:

- **Hierarchical Reinforcement Learning (HRL-MADDPG) [51]:** Based on the Multi-Agent Deep Deterministic Policy Gradient algorithm, it uses a hierarchical structure to handle complex mission decomposition. This was selected as a representative state-of-the-art method from the data-driven, multi-agent reinforcement learning (MARL) domain.
- **Rule-Based Coordination (RBC) [52]:** Employs pre-defined mission decomposition rules and coordination strategies with static task assignment and path planning algorithms. This was chosen to represent a classic, knowledge-driven but non-cognitive approach.

For a fair comparison, the hyperparameters for HRL-MADDPG were tuned using a grid search to achieve optimal performance in our simulation environment, and the rules for RBC were designed by domain experts to be as effective as possible.

### 5.3. Performance Analysis

To quantitatively assess the effectiveness of the proposed CogMUS framework, we conducted a comprehensive comparison against HRL-MADDPG and RBC in a highly dynamic simulation environment. The evaluation focused on three core metrics: TUA, CE, and TCR. Experiments were run for 400 epochs in each of the five typical cooperative mission scenarios, with the results meticulously recorded and analyzed. To ensure the statistical reliability of our final performance comparison, the evaluation for each method across all five scenarios was repeated 30 times with different random seeds for target and obstacle placement. The results presented in the figures and tables represent the mean performance across these independent runs.

Figure 4 illustrates the evolution of TUA during training. The results clearly indicate that CogMUS demonstrates significantly superior learning efficiency and final performance. Our method exhibits an exceptionally steep initial learning curve, surpassing 80% accuracy within approximately 120 training epochs and eventually converging to a stable, high level of around 82.5% after 250 epochs. This exceptional performance is primarily attributed to its knowledge-driven cognitive mechanism. The Soar architecture leverages the rich set of production rules in its LTM to perform efficient symbolic reasoning and zero-shot decomposition of high-level mission intent, thereby bypassing the extensive trial-and-error exploration required by reinforcement learning.

Figure 5 depicts the change in CE over training epochs, a metric that directly reflects the swarm's ability to translate mission understanding into spatiotemporally coordinated actions. CogMUS again shows the best performance, with its CE rapidly climbing and stabilizing at an extremely high level of over 90% with minimal volatility. This is attributed to two core functionalities within the CogMUS framework: (1) precise mission understanding and decomposition lay a clear foundation for subsequent coordination; (2) Soar agents can achieve deep intent alignment and proactive trajectory planning through shared WM states and built-in deconfliction rules (as shown in Rule Example 2, Section 4.2), thus minimizing delays, conflicts, and resource wastage. This result indicates that CogMUS not only "understands" the mission but also effectively translates this high-level understanding into efficient group actions.

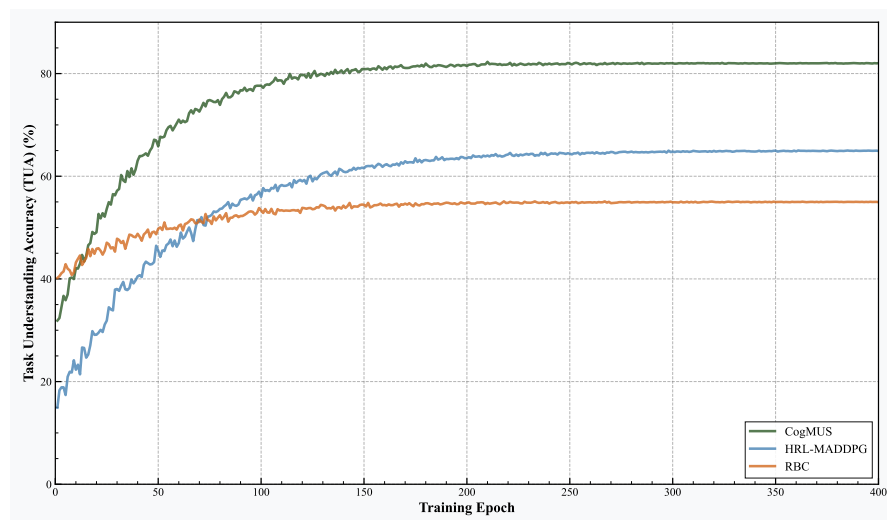


**Figure 4.** Comparison of TUA during training for the three methods.

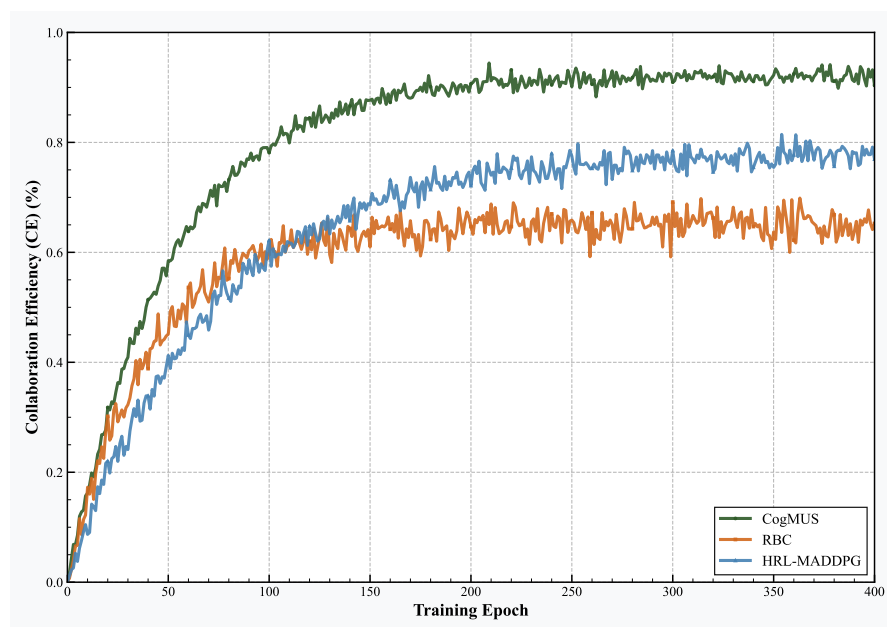


**Figure 5.** Comparison of CE during training for the three methods.

To evaluate the framework's adaptability and robustness, Figure 6 presents the average TCR of the three methods across five different mission scenarios. The bar chart intuitively shows that CogMUS achieved the highest completion rate in all tested scenarios, with an average TCR of 84.6%, significantly higher than HRL-MADDPG (71.1%) and RBC (75.2%). Its advantage is particularly pronounced in the "Cooperative Defense" scenario, which demands rapid response to dynamic anomalies and agile resource scheduling, where its TCR approaches 90%. This validates the framework's strong generalization capability and adaptability to different mission constraints. In contrast, HRL-MADDPG's performance fluctuates considerably across scenarios, suggesting potential overfitting to its training environment, while RBC performs more consistently, its overall performance ceiling is markedly lower than that of CogMUS, rendering it incapable of handling all types of complex missions. This series of comparative results substantiates the superior execution capability and high reliability of the CogMUS framework under diverse operational demands.

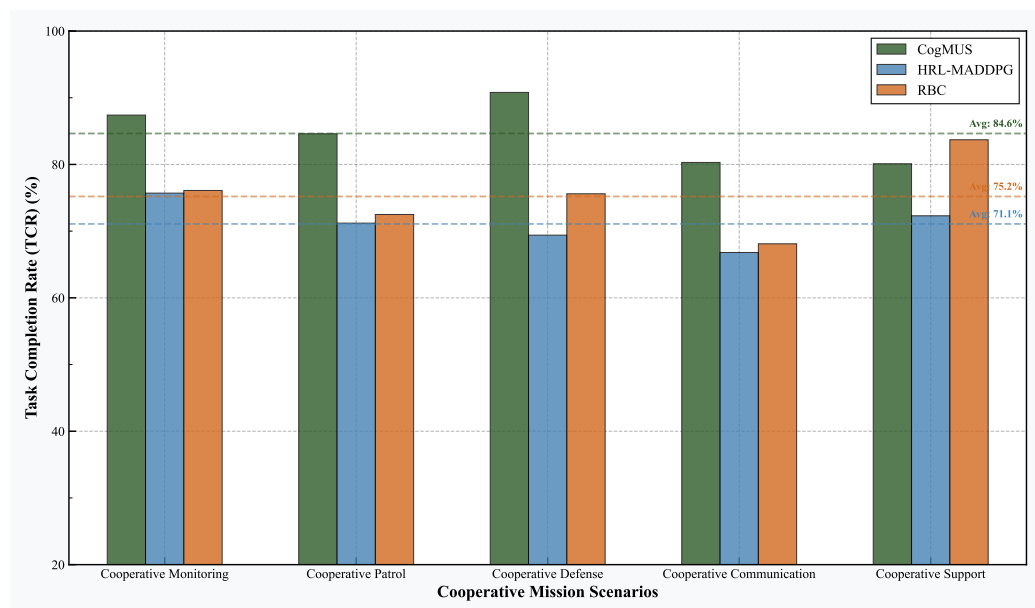


**Figure 6.** Comparison of TCR across five typical cooperative mission scenarios.

Table 3 summarizes the average performance metrics for the three methods across all experiments. The data clearly indicate that CogMUS comprehensively and significantly outperforms both baseline methods across all three key metrics.

**Table 3.** Overall performance evaluation results (mean  $\pm$  standard deviation).

| Metric  | CogMUS         | HRL-MADDPG     | RBC            |
|---------|----------------|----------------|----------------|
| TUA (%) | 82.5 $\pm$ 2.8 | 55.8 $\pm$ 4.5 | 65.2 $\pm$ 3.3 |
| CE (%)  | 90.3 $\pm$ 3.1 | 77.6 $\pm$ 5.2 | 64.6 $\pm$ 4.1 |
| TCR (%) | 84.6 $\pm$ 3.5 | 71.1 $\pm$ 6.1 | 75.2 $\pm$ 4.9 |

Based on the above experimental results, compared to the rule-based RBC method, learning-based approaches like CogMUS and HRL-MADDPG demonstrate stronger adaptability and generalization by continuously optimizing their policies through interaction with the environment. Due to its hybrid symbolic-subsymbolic nature, CogMUS particularly excels in learning efficiency and knowledge utilization. Furthermore, its faster convergence speed compared to HRL-MADDPG alleviates the burden of high training costs.

#### 5.4. Ablation Study

To validate the contribution of specific knowledge components, we conducted an ablation study in the cooperative defense scenario. We created a variant of our system, “CogMUS (No Deconfliction),” in which we removed all production rules from LTM related to proactive 3D airspace deconfliction (such as Rule Example 2 in Section 4.2). We measured the number of safety violations (near-miss incidents) and the impact on CE and TCR.

The results, presented in Table 4, show that removing the deconfliction module led to a cascading degradation in system performance. The most direct impact was a surge in safety violations from an average of 1.2 to 27.8 per mission, confirming the module’s central role in preventing trajectory conflicts. This reduction in safety translated directly into a loss of operational efficiency. Frequent reactive avoidance maneuvers forced UAVs to deviate from their optimal routes, causing CE to drop significantly from 91.5% to 75.3%. More critically, these unplanned deviations and the resulting desynchronization of coordinated actions directly led to mission failures, ultimately reflected in a substantial decrease in TCR from

89.3% to 78.5%. This study clearly demonstrates that the proposed proactive deconfliction mechanism is not merely a safety feature but a critical component for maintaining high mission execution efficiency and ensuring mission success in dense multi-UAV airspace.

**Table 4.** Ablation study results.

| Methods                | Safety Violations (avg) | TCR (%) | CE (%) |
|------------------------|-------------------------|---------|--------|
| CogMUS (Full)          | 1.2                     | 89.3    | 91.5   |
| CogMUS (No Deconflict) | 27.8                    | 78.5    | 75.3   |

## 6. Discussion

### 6.1. Interpretation of Key Findings

The experimental results strongly support our central hypothesis: a cognitive architecture-based approach offers a robust and effective solution for multi-UAV mission understanding. The superiority of CogMUS over the state-of-the-art MARL method (HRL-MADDPG) can be attributed to several factors. Soar’s ability to leverage a rich symbolic knowledge base enables it to perform zero-shot reasoning on new problems, whereas HRL requires extensive training for each new mission type and struggles with generalization. The explicit, rule-based reasoning in Soar provides transparency and interpretability, which are critical features for high-stakes or safety-critical applications. Compared to the rigid RBC method, Soar’s dynamic DC allows it to adapt rapidly to unforeseen events, a capability strongly evidenced in the robustness tests. From a cognitive perspective, the three compared approaches embody different levels of intelligence. The RBC method represents a purely reactive system. The HRL-MADDPG method demonstrates adaptive learning. CogMUS, in contrast, operates at a higher cognitive level, utilizing explicit, symbolic representations of goals, tasks, and intentions, representing a form of deliberate, goal-directed reasoning that is closer to human problem-solving.

### 6.2. Implications for Autonomous UAV System Design

This research has significant implications for the design of next-generation autonomous systems. It suggests a trend away from purely end-to-end learning models towards hybrid architectures that combine the reasoning power of symbolic AI with the adaptive strengths of machine learning. The success of our distributed cognitive model highlights the importance of designing explicit communication and coordination protocols that allow agents to share not only data but also their intent and reasoning processes, thereby enabling a deeper level of collective intelligence. A key aspect of this design is its inherent robustness to environmental uncertainty. Unlike traditional planners that may fail when pre-conditions are not met, the Soar-based agents in CogMUS continuously operate within their decision cycle. When an unexpected event occurs, the agent’s WM is immediately updated. This change in WM can invalidate the premises of the current plan and trigger new production rules that propose alternative operators. This inherent reactivity allows the swarm to gracefully handle uncertainty without requiring an exhaustive pre-computation of all possible contingencies.

### 6.3. Limitations and Future Work

This study is not without limitations. First, the knowledge base (LTM rules) in CogMUS was manually engineered by domain experts. This is a time-consuming process that may not cover all possible contingencies. A key area for future work is the integration of machine learning techniques to automatically acquire and refine these rules from experience. Second, while the simulation environment was complex, it does not fully capture

real-world noise and uncertainty (i.e., the “sim-to-real” gap). The logical next steps are to validate the CogMUS framework in higher-fidelity robotics simulators (e.g., Gazebo with ROS2) and eventually on physical UAV hardware platforms. Finally, we plan to extend the framework to support more sophisticated human-in-the-loop collaboration, allowing human commanders to interact with the UAV swarm at a high cognitive level to provide guidance and dynamically adjust mission priorities. The symbolic nature of Soar’s reasoning also makes it highly interpretable, which could enable transparent oversight and build trust in the autonomous system.

## 7. Conclusions

This paper addressed the formidable challenge of mission understanding for multi-UAV systems operating in complex and dynamic cooperative operational environments by proposing CogMUS, a novel framework based on the Soar cognitive architecture. By equipping each UAV with a cognitive agent capable of human-like reasoning, our system effectively interprets high-level mission intent, decomposes complex tasks, and coordinates actions among multiple agents in a distributed and robust manner. This enables effective mission decomposition, intent alignment among UAVs, and the detection and resolution of potential conflicts. Through extensive simulation experiments, we have demonstrated the superiority of our approach over reinforcement learning and traditional rule-based baselines, achieving significant improvements in TUA, CE, and TCR. This work highlights the significant potential of cognitive architectures in bridging the gap between abstract human commands and autonomous multi-agent execution, paving the way for more intelligent, adaptive, and trustworthy multi-UAV systems for future complex operations. Future work will focus on human–machine collaboration and large-scale swarm mission understanding, promising to further advance the intelligence and collaborative effectiveness of multi-UAV swarm.

**Author Contributions:** Conceptualization, J.H. and T.W.; methodology, J.H.; software, J.H.; validation, T.W.; formal analysis, J.H.; investigation, J.H.; resources, H.W.; data curation, J.C.; writing—original draft preparation, J.H.; writing—review and editing, J.H.; visualization, H.W. and J.C.; supervision, T.W.; project administration, J.H.; funding acquisition, T.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors were supported by the National Basic Research Program of China (Grant No. 2022-JCJQ-JJ-0683).

**Data Availability Statement:** The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

**DURC Statement:** The current research is limited to the academic field of UAV autonomous systems and cognitive robotics, which is beneficial for enhancing the efficiency, safety and autonomy of multi-UAV in civilian and defensive applications, such as disaster response, infrastructure inspection, and environmental monitoring. This research does not pose a threat to public health or national security. The authors acknowledge the dual-use potential of the research involving cognitive framework for multi-UAV mission understanding and confirm that all necessary precautions have been taken to prevent potential misuse. As an ethical responsibility, the authors strictly adhere to relevant national and international laws about dual-use research of concern (DURC). The authors advocate for responsible deployment, ethical considerations, regulatory compliance, and transparent reporting to mitigate misuse risks and foster beneficial outcomes.

**Acknowledgments:** The authors appreciate the editors and anonymous reviewers for their valuable recommendations.

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

## References

1. Zhang, H.; Li, W.; Zheng, J.; Liu, H.; Zhang, P.; Peng, G.; Gan, X. Manned/unmanned aerial vehicle cooperative combat system: Concepts, technologies, and challenges. *Acta Aeronaut. Astronaut. Sin.* **2024**, *45*, 029653.
2. Mohsan, S.A.H.; Khan, M.A.; Noor, F.; Ullah, I.; Alsharif, M.H. Towards the unmanned aerial vehicles (UAVs): A comprehensive review. *Drones* **2022**, *6*, 147. [\[CrossRef\]](#)
3. Wang, X.; Cao, Y.; Ding, M.; Wang, X.; Yu, W.; Guo, B. Research Progress in Modeling and Evaluation of Cooperative Operation System-of-Systems for Manned–Unmanned Aerial Vehicles. *IEEE Aerosp. Electron. Syst. Mag.* **2023**, *39*, 6–31. [\[CrossRef\]](#)
4. Ciolponea, C.A. The Integration of Unmanned Aircraft System (UAS) in Current Combat Operations. *Land Forces Acad. Rev.* **2022**, *27*, 108. [\[CrossRef\]](#)
5. Jie, C.; Bin, X. Key scientific issues in autonomous collaboration of manned/unmanned systems. *Chin. Sci. Inf. Sci.* **2018**, *48*, 1270–1274.
6. Shu, Z.; Wang, W.; Wang, R. Design of an optimized architecture for manned and unmanned combat system-of-systems: Formulation and coevolutionary optimization. *IEEE Access* **2018**, *6*, 52725–52740. [\[CrossRef\]](#)
7. Kolling, A.; Walker, P.; Chakraborty, N.; Sycara, K.; Lewis, M. Human Interaction with Robot Swarms: A Survey. *IEEE Trans.-Hum.-Mach. Syst.* **2016**, *46*, 9–26. [\[CrossRef\]](#)
8. Zhang, X.; Wang, W.; Ren, S.; Gong, X.; Yang, Y.; Wang, J. A Two-Phase Task Allocation Strategy With a Hybrid Architecture. In Proceedings of the 2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Tianjin, China, 8–10 May 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 55–60.
9. Ruan, T.; Ramesh, A.; Wang, H.; Johnstone-Morfoisse, A.; Altindal, G.; Norman, P.; Nikolaou, G.; Stolkin, R.; Chiou, M. A Framework for Semantics-based Situational Awareness during Mobile Robot Deployments. *arXiv* **2025**, arXiv:2502.13677. [\[CrossRef\]](#)
10. Zhang, S.; Lin, X.; Li, X.; Ren, A. Service robots' anthropomorphism: Dimensions, factors and internal relationships. *Electron. Mark.* **2022**, *32*, 277–295. [\[CrossRef\]](#)
11. Qian, C.; He, B.; Zhuang, Z.; Deng, J.; Qin, Y.; Cong, X.; Zhang, Z.; Zhou, J.; Lin, Y.; Liu, Z.; et al. Tell me more! towards implicit user intention understanding of language model driven agents. *arXiv* **2024**, arXiv:2402.09205. [\[CrossRef\]](#)
12. Cui, Y.; Liang, Y.; Luo, Q.; Shu, Z.; Huang, T. Resilient consensus control of heterogeneous multi-UAV systems with leader of unknown input against Byzantine attacks. *IEEE Trans. Autom. Sci. Eng.* **2024**, *22*, 5388–5399. [\[CrossRef\]](#)
13. Telli, K.; Kraa, O.; Himeur, Y.; Ouamane, A.; Boumehraz, M.; Atalla, S.; Mansoor, W. A comprehensive review of recent research trends on unmanned aerial vehicles (uavs). *Systems* **2023**, *11*, 400. [\[CrossRef\]](#)
14. Ma, Z.; Gong, H.; Xiong, J.; Wang, X. Heterogeneous Multi-Agent Task Allocation Based on Graph-Based Convolutional Assignment Neural Network. *IEEE Internet Things J.* **2025**, *12*, 17281–17299. [\[CrossRef\]](#)
15. Hamissi, A.; Dhraief, A.; Sliman, L. A comprehensive survey on conflict detection and resolution in unmanned aircraft system traffic management. *IEEE Trans. Intell. Transp. Syst.* **2024**, *26*, 1395–1418. [\[CrossRef\]](#)
16. Laird, J.E. *The Soar Cognitive Architecture*; MIT Press: Cambridge, MA, USA, 2019.
17. Guo, Y.; Ren, Z.; Wang, C. iMTSP: Solving min-max multiple traveling salesman problem with imperative learning. In Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 10245–10252.
18. Konstantakopoulos, G.D.; Gayialis, S.P.; Kechagias, E.P. Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification. *Oper. Res.* **2022**, *22*, 2033–2062. [\[CrossRef\]](#)
19. Ait Saadi, A.; Soukane, A.; Meraihi, Y.; Benmessaoud Gabis, A.; Mirjalili, S.; Ramdane-Cherif, A. UAV path planning using optimization approaches: A survey. *Arch. Comput. Methods Eng.* **2022**, *29*, 4233–4284. [\[CrossRef\]](#)
20. Albert, A.; Leira, F.S.; Imsland, L.S. UAV path planning using MILP with experiments. *Model. Identif. Control* **2017**, *38*, 21–32. [\[CrossRef\]](#)
21. Quinton, F.; Grand, C.; Lesire, C. Market approaches to the multi-robot task allocation problem: A survey. *J. Intell. Robot. Syst.* **2023**, *107*, 29. [\[CrossRef\]](#)
22. Hooshyar, M.; Huang, Y.M. Meta-heuristic algorithms in UAV path planning optimization: A systematic review (2018–2022). *Drones* **2023**, *7*, 687. [\[CrossRef\]](#)
23. Du, W.; Ding, S. A survey on multi-agent deep reinforcement learning: From the perspective of challenges and applications. *Artif. Intell. Rev.* **2021**, *54*, 3215–3238. [\[CrossRef\]](#)
24. Wu, J.; Li, D.; Yu, Y.; Gao, L.; Wu, J.; Han, G. An attention mechanism and adaptive accuracy triple-dependent MADDPG formation control method for hybrid UAVs. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 11648–11663. [\[CrossRef\]](#)
25. Nguyen, K.; Dang, V.T.; Pham, D.D.; Dao, P.N. Formation control scheme with reinforcement learning strategy for a group of multiple surface vehicles. *Int. J. Robust Nonlinear Control* **2024**, *34*, 2252–2279. [\[CrossRef\]](#)
26. Gronauer, S.; Diepold, K. Multi-agent deep reinforcement learning: A survey. *Artif. Intell. Rev.* **2022**, *55*, 895–943. [\[CrossRef\]](#)

27. Hannoun, M.; Boissier, O.; Sichman, J.S.; Sayettat, C. MOISE: An organizational model for multi-agent systems. In Proceedings of the Ibero-American Conference on Artificial Intelligence, Atibaia, SP, Brazil, 19–22 November 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 156–165.
28. Boissier, O.; Bordini, R.H.; Hubner, J.; Ricci, A. *Multi-Agent Oriented Programming: Programming Multi-Agent Systems Using JaCaMo*; Mit Press: Cambridge, MA, USA, 2020.
29. Li, P.; An, Z.; Abrar, S.; Zhou, L. Large language models for multi-robot systems: A survey. *arXiv* **2025**, arXiv:2502.03814. [[CrossRef](#)]
30. Chen, D.; Mooney, R. Learning to interpret natural language navigation instructions from observations. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 7–11 August 2011; Volume 25, pp. 859–865.
31. Javaid, S.; Fahim, H.; He, B.; Saeed, N. Large language models for uavs: Current state and pathways to the future. *IEEE Open J. Veh. Technol.* **2024**, *5*, 1166–1192. [[CrossRef](#)]
32. Baxter, D.P.; Hepworth, A.J.; Joiner, K.F.; Abbass, H. On the premise of a swarm guidance ontology for human-swarm teaming. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Atlanta, GA, USA, 10–14 October 2022; SAGE Publications Sage CA: Los Angeles, CA, USA, 2022; Volume 66, pp. 2249–2253.
33. Mishra, N.; Kumaraswamy, B. Deployment of Ontologies as Knowledge Reasoning Technique for Making Autonomous Robots: A Survey. In Proceedings of the 2025 International Conference on Intelligent Control, Computing and Communications (IC3), Mathura, India, 13–14 February 2025; IEEE: Piscataway, NJ, USA, 2025; pp. 943–947.
34. Brown, D.S.; Niekum, S. Machine teaching for inverse reinforcement learning: Algorithms and applications. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7749–7758.
35. Ravichandar, H.; Polydoros, A.S.; Chernova, S.; Billard, A. Recent advances in robot learning from demonstration. *Annu. Rev. Control. Robot. Auton. Syst.* **2020**, *3*, 297–330. [[CrossRef](#)]
36. Xi, X.; Zhu, S. A comprehensive review of task understanding of command-triggered execution of tasks for service robots. *Artif. Intell. Rev.* **2023**, *56*, 7137–7193. [[CrossRef](#)]
37. Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. A survey on large language model based autonomous agents. *Front. Comput. Sci.* **2024**, *18*, 186345. [[CrossRef](#)]
38. Mehandru, N.; Miao, B.Y.; Almaraz, E.R.; Sushil, M.; Butte, A.J.; Alaa, A. Evaluating large language models as agents in the clinic. *NPJ Digit. Med.* **2024**, *7*, 84. [[CrossRef](#)]
39. Xi, Z.; Ding, Y.; Chen, W.; Hong, B.; Guo, H.; Wang, J.; Yang, D.; Liao, C.; Guo, X.; He, W.; et al. Agentgym: Evolving large language model-based agents across diverse environments. *arXiv* **2024**, arXiv:2406.04151.
40. Zahedifar, R.; Mirghasemi, S.A.; Baghshah, M.S.; Taheri, A. LLM-Agent-Controller: A Universal Multi-Agent Large Language Model System as a Control Engineer. *arXiv* **2025**, arXiv:2505.19567.
41. Zhang, Z.; Zhang-Li, D.; Yu, J.; Gong, L.; Zhou, J.; Hao, Z.; Jiang, J.; Cao, J.; Liu, H.; Liu, Z.; et al. Simulating classroom education with llm-empowered agents. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Albuquerque, NM, USA, 29 April–4 May 2025; pp. 10364–10379.
42. Ritter, F.E.; Tehranchi, F.; Oury, J.D. ACT-R: A cognitive architecture for modeling cognition. *Wiley Interdiscip. Rev. Cogn. Sci.* **2019**, *10*, e1488. [[CrossRef](#)]
43. Saha, O.; Dasgupta, P. A comprehensive survey of recent trends in cloud robotics architectures and applications. *Robotics* **2018**, *7*, 47. [[CrossRef](#)]
44. Moulin-Frier, C.; Fischer, T.; Petit, M.; Pointeau, G.; Puigbo, J.Y.; Pattacini, U.; Low, S.C.; Camilleri, D.; Nguyen, P.; Hoffmann, M.; et al. DAC-h3: A proactive robot cognitive architecture to acquire and express knowledge about the world and the self. *IEEE Trans. Cogn. Dev. Syst.* **2017**, *10*, 1005–1022. [[CrossRef](#)]
45. Pinto, M.F.; Honório, L.M.; Marcato, A.L.; Dantas, M.A.; Melo, A.G.; Capretz, M.; Urdiales, C. Arcog: An aerial robotics cognitive architecture. *Robotica* **2021**, *39*, 483–502. [[CrossRef](#)]
46. Emel'yanov, S.; Makarov, D.; Panov, A.I.; Yakovlev, K. Multilayer cognitive architecture for UAV control. *Cogn. Syst. Res.* **2016**, *39*, 58–72. [[CrossRef](#)]
47. Luo, F.; Zhou, Q.; Fuentes, J.; Ding, W.; Gu, C. A soar-based space exploration algorithm for mobile robots. *Entropy* **2022**, *24*, 426. [[CrossRef](#)]
48. Gunetti, P.; Thompson, H.; Dodd, T. Autonomous mission management for UAVs using soar intelligent agents. *Int. J. Syst. Sci.* **2013**, *44*, 831–852. [[CrossRef](#)]
49. van Ments, L.; Treur, J. Reflections on dynamics, adaptation and control: A cognitive architecture for mental models. *Cogn. Syst. Res.* **2021**, *70*, 1–9. [[CrossRef](#)]
50. Ramos, G.S.; Henriques, F.D.R.; Haddad, D.B.; Andrade, F.A.; Pinto, M.F. ARCog-NET: An aerial robot cognitive network architecture for swarm applications development. *IEEE Access* **2024**, *12*, 129040–129063. [[CrossRef](#)]

51. Hutsebaut-Buysse, M.; Mets, K.; Latré, S. Hierarchical reinforcement learning: A survey and open research challenges. *Mach. Learn. Knowl. Extr.* **2022**, *4*, 172–221. [[CrossRef](#)]
52. Von Riegen, M.; Husemann, M.; Fink, S.; Ritter, N. Rule-based coordination of distributed web service transactions. *IEEE Trans. Serv. Comput.* **2009**, *3*, 60–72. [[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.