
An LLM-based Framework for Human-Swarm Teaming Cognition in Disaster Search and Rescue

Kailun Ji^{1*} Xiaoyu Hu^{1*} Xinyu Zhang^{1,2†} Jun Chen^{1,2}

¹School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China
²Chongqing Institute for Brain and Intelligence, Guangyang Bay Laboratory, Chongqing 400064, China

Abstract

Large-scale disaster Search And Rescue (SAR) operations are persistently challenged by complex terrain and disrupted communications. While Unmanned Aerial Vehicle (UAV) swarms offer a promising solution for tasks like wide-area search and supply delivery, yet their effective coordination places a significant cognitive burden on human operators. The core human-machine collaboration bottleneck lies in the “intention-to-action gap”, which is an error-prone process of translating a high-level rescue objective into a low-level swarm command under high intensity and pressure. To bridge this gap, this study proposes a novel LLM-CRF system that leverages Large Language Models (LLMs) to model and augment human-swarm teaming cognition. The proposed framework initially captures the operator’s intention through natural and multi-modal interactions with the device via voice or graphical annotations. It then employs the LLM as a cognitive engine to perform intention comprehension, hierarchical task decomposition, and mission planning for the UAV swarm. This closed-loop framework enables the swarm to act as a proactive partner, providing active feedback in real-time while reducing the need for manual monitoring and control, which considerably advances the efficacy of the SAR task. We evaluate the proposed framework in a simulated SAR scenario. Experimental results demonstrate that, compared to traditional order and command-based interfaces, the proposed LLM-driven approach reduced task completion time by approximately 64.2% and improved task success rate by 7%. It also leads to a considerable reduction in subjective cognitive workload, with NASA-TLX scores dropping by 42.9%. This work establishes the potential of LLMs to create more intuitive and effective human-swarm collaborations in high-stakes scenarios.

1 Introduction

In large-scale disaster scenarios, such as earthquakes, floods, and fires, to name a few, securing the “golden 72-hour” rescue window is paramount for saving lives and reducing losses [1, 2, 3]. Under such a condition, Unmanned Aerial Vehicle (UAV) swarms have emerged as a critical asset in this race against time. They are capable of rapid deployment to high-risk and inaccessible areas, and they can collaboratively perform essential tasks, including wide-area search [4, 5], target identification [6], building damage assessment [7, 8], and emergency medical supply delivery [9]. The powerful capabilities of drone swarms, however, introduce a significant operational bottleneck in the form of an immense cognitive workload for human operators. This difficulty is further compounded by the need to process and fuse multi-source and heterogeneous information streams, such as real-time drone video feedback, infrared thermal imaging, Geographic Information System (GIS) map data,

*Equal contribution.

†Corresponding author: Xinyu.Zhang@nwpu.edu.cn

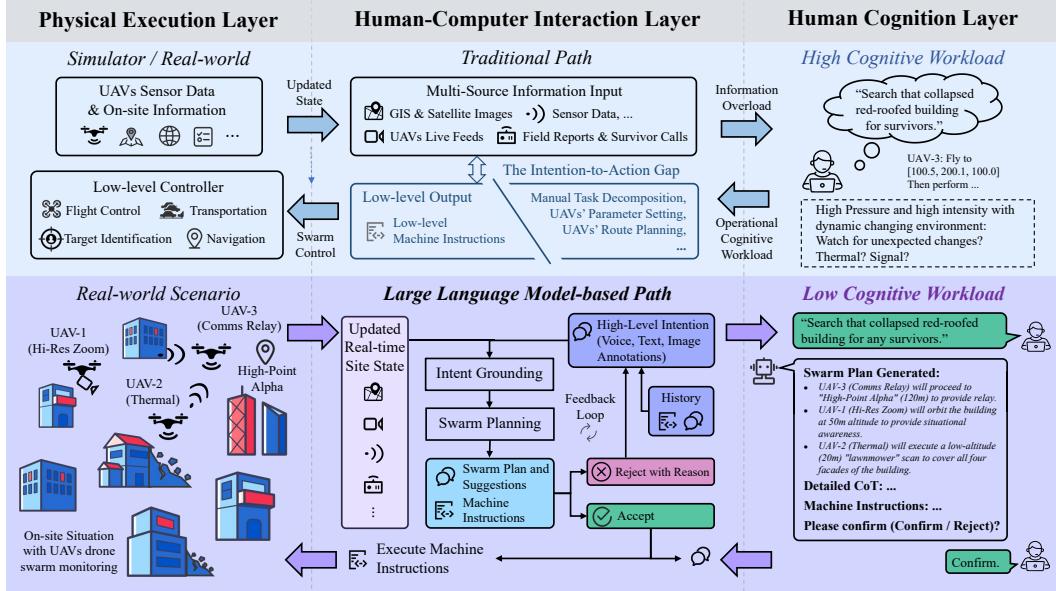


Figure 1: **The UAV Swarm Disaster SAR Workflow.** The traditional approach (above) creates a significant “intention-to-action gap”, imposing a heavy cognitive workload on human operators. Our proposed framework (below) bridges this gap by leveraging an LLM-based core to intelligently decompose high-level multi-modal intention into an executable swarm plan.

and survivor reports, which is a demanding task that requires operators to sustain a high level of situational awareness throughout the entire operation [10].

Traditional human-swarm interaction, which operates on a “command-response” paradigm, exacerbates the operator’s cognitive workload [11, 12]. This approach generally requires the operator to manually decompose high-level Search And Rescue (SAR) intentions into a lengthy series of low-level machine instructions. For instance, to execute an order such as “immediately send two drones to the collapsed red-roofed building in area B to check for life signals, and have another drone provide a high-altitude communication relay”. An operator must manually: 1) identify the building’s precise coordinates, 2) plan individual obstacle avoidance routes for the two drones, 3) configure their sensor payloads (e.g., activating thermal imaging), 4) set loitering waypoints and altitude for the relay drone, and 5) continuously monitor and intervene the devices in real-time. This manual “intention-to-command” translation is highly inefficient and error-prone under high pressure and high intensity tasks, creating a significant gap between human decision-making and machine execution.

In addition, the process of translating human intention into commands is significantly influenced by both operator’s preferences and mission-specific requirements, which directly shape the desired swarm behavior. For example, a wide-area search task prioritizes coverage efficiency, favoring a Z-shaped scanning pattern. Conversely, a building damage assessment demands high precision, for which a concentrated orbiting pattern is better suited. Traditional systems are inherently inflexible and cannot dynamically incorporate such implicit contextual knowledge or user-specific preferences, which limits their overall effectiveness and adaptability [13].

Against this backdrop, Large Language Models (LLMs) demonstrate immense potential for addressing these bottlenecks [14, 15, 16]. The powerful capabilities of LLMs in natural language understanding, contextual reasoning [17, 18, 19], multi-turn dialogue, and robust inference, have led to their effective applications in various complex human-machine collaboration scenarios, such as robotics (e.g., Google’s SayCan)[20], code generation (e.g., Github Copilot), and complex data analysis (e.g., OpenAI’s Code Interpreter). The emergence of LLMs makes it possible for machines to comprehend high-level and ambiguous instructions. Therefore, this paper proposes a novel LLM-based cognitive reasoning framework (LLM-CRF) for human-swarm teaming. The proposed framework leverages the natural language understanding and reasoning capabilities of an LLM to model and augment the operator’s cognitive processes. It captures high-level intention through multi-modal interactions (e.g.,

voice, gestures) [21, 22, 23] and employs the LLM as a central cognitive engine to autonomously perform intention comprehension, hierarchical task decomposition, and mission planning for the UAV swarm. By effectively closing the loop between human intention and swarm action, this approach transforms the UAV swarm from a passive tool into a proactive partner, thereby significantly advancing the efficacy of SAR operations. The gap between the traditional and the proposed approaches is demonstrated in Figure 1.

2 Related works

Traditional research on UAV swarm coordination has predominantly addressed the algorithmic challenges of Multi-Agent Task Allocation (MATA) and Multi-Agent Path Planning (MAPP) [11, 12, 24]. This substantial body of research has yielded computationally efficient methods, including heuristic algorithms and market-based mechanisms to optimize swarm behavior for predefined objectives [25, 26, 27, 28]. However, these frameworks share a fundamental limitation in their underlying assumption, that a human operator can formally and precisely articulate mission goals, constraints, and cost functions in a structured, machine-readable format. This critical prerequisite, the “intent-to-command” translation, imposes a substantial cognitive burden on the operator [10, 23]. Consequently, the human capability is effectively reduced from a strategic commander to a low-level programmer, which is a role mismatch that proves particularly debilitating in the dynamic, high-stress environments characteristic of disaster response.

To mitigate such a burden, prior research has explored more intuitive interaction modalities, such as voice and gesture control [29]. These systems typically employ conventional Natural Language Processing (NLP) techniques, including semantic parsing and intention classification [30], to map a constrained vocabulary of predefined commands (e.g., “take off”, “scan area”) onto specific robotic functionalities. While representing a step forward, these approaches are inherently brittle and lack robustness, as they fail to comprehend the complex, contextual, and ambiguous instructions that typify real-world mission directives, such as “check that collapsed red-roofed building for survivors”. While the advent of LLMs [14, 15, 31] and Vision-Language Models (VLMs) [32, 33, 34] has provided a transformative new path. Their powerful common-sense reasoning, in-context learning, and planning capabilities [17] have acted as the “brain” for embodied agents. Building on their success in robotics, LLMs have shown the capacity to ground high-level instructions into executable action sequences for manipulation and navigation [35, 20], and this paradigm is now being extended to UAVs. Recent research confirms that LLMs can effectively generate navigation waypoints, write flight control scripts, and perform high-level task planning for drone operations [36, 37].

Despite recent advances, current methods are not well-suited for disaster response, revealing several key shortcomings. A major limitation is that existing works overwhelmingly focus on single-agent control [35, 20], which does not address the one-to-many decomposition for swarm coordination. This type of approach lacks the ability of complex resource allocation, role assignment, and spatio-temporal de-confliction. In addition, these systems are generally performed and evaluated in simple, structured, and simulated environments, which do not reflect the dynamic, unpredictable, and communication-constrained nature of a disaster site. Furthermore, the safety-critical context of SAR cannot tolerate the known risk of LLM hallucinations [38, 39]. Blindly executing a factually incorrect or made-up plan in a rescue mission is unacceptable, yet current frameworks lack the robust verification and human-in-the-loop feedback mechanisms required for such high-stakes operations. Hence, this work aims to tackle these concerns by creating a framework that can reliably and safely translate human commands into coordinated actions for drone swarms in real-world SAR missions.

3 Method

This paper introduces a novel LLM-based Cognitive Reasoning Framework (LLM-CRF) to bridge the “intention-to-action” gap in UAV swarm control, translating high-level human intent into executable robotic actions. It functions as a cognitive engine between a front-end multi-modal interface and a back-end UAV action library, as illustrated in Figure 2.

The LLM-CRF engine operates on a hierarchical, multi-model architecture. At its core, an LLM functions as the central reasoning agent, which is supported by a suite of specialized perception and transcription tools (i.e., Qwen-14B-Chat [40]). Specifically, a Vision-Language Model (VLM)

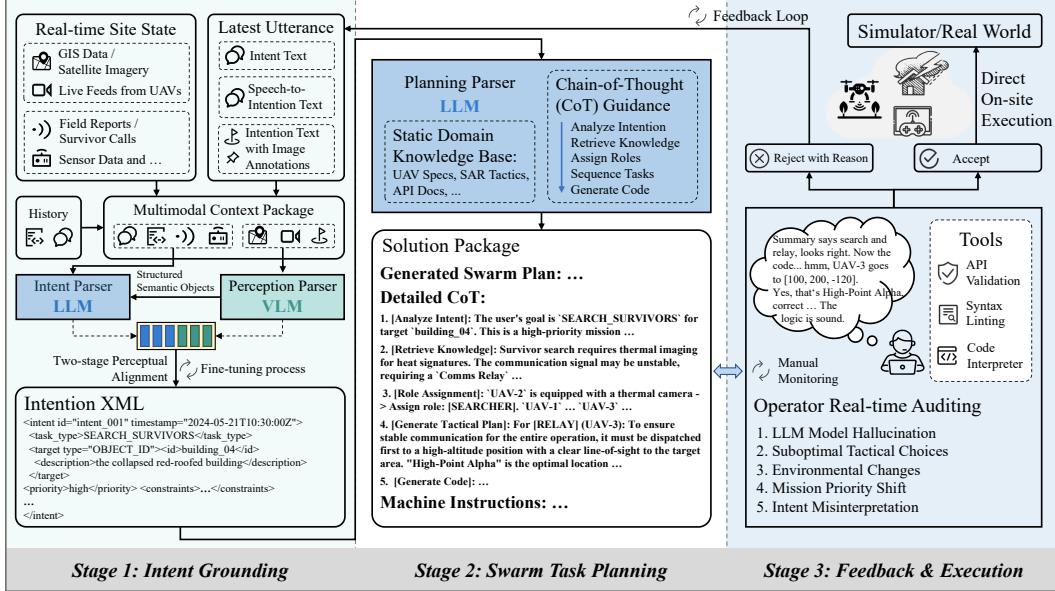


Figure 2: **The proposed LLM-based Cognitive Reasoning Framework (LLM-CRF).** The system translates raw multi-modal inputs into executable actions through a three-stage process, including intent grounding, swarm task planning, and feedback and execution.

(i.e., LLaVA-1.6 [41]) is utilized as a dedicated visual perception module, while Whisper [42] serves as the speech-to-text module. Our methodology focuses on the deep, domain-specific adaptation of these models to form a cohesive expert system. The system initiates each decision cycle by constructing a multi-model context package. This package integrates all pertinent information for reasoning, including the operator’s *latest_utterance*, associated *image_annotations*, the complete *dialogue_history*, and the *world_state*, describing the global site situation and UAV statuses. This consolidated package provides a unified input for all subsequent processing stages.

3.1 Intent Grounding via Perceptual Alignment

The initial stage of the LLM-CRF is Intent Grounding, which converts an operator’s raw, multi-modal inputs into a structured and machine-executable representation. This requires the LLM to achieve a contextualized understanding of the disaster scene, semantically grounding the operator’s linguistic commands within the UAVs’ visual perceptions of the environment. To this end, we designed a two-stage Perceptual Alignment Fine-tuning process to address the domain shift problem when applying general-purpose VLMs to the specialized domain of UAV-based disaster response. The fine-tuning is applied to achieve precise alignment between the VLM’s representations and the LLM’s semantic space, adapting it for the UAV-SAR domain through the following specialized stages:

- Stage 1 - Vision-language feature alignment pre-training: This initial stage aims to establish a foundational mapping between visual features and general linguistic concepts. During this stage, we freeze the pre-trained vision encoder $f_{vision}(\cdot)$ and LLM $f_{lm}(\cdot)$, training only a lightweight adapter module $f_{adapter}(\cdot; \theta_{adapter})$. This adapter is trained on large-scale aerial image-text pairs (I, T) (e.g., from RS5M [43]) to learn a projection that effectively maps visual features from the aerial domain into the LLM’s embedding space. The optimization objective is to minimize the contrastive loss between the projected visual features and the text embeddings:

$$\mathcal{L}_{stage1} = \mathcal{L}_{contrastive}(f_{adapter}(f_{vision}(I)), f_{lm}(T)) \quad (1)$$

This provides the model with a preliminary and generic understanding of visual semantics.

- Stage 2 - Domain-specific multi-modal instruction fine-tuning: This stage elevates the model from a passive observer to an active perceptual agent within the disaster response context. Building upon the first stage, we continue to keep the vision encoder frozen while performing parameter-efficient fine-tuning via Low-Rank Adaptation (LoRA) [37]), on both the adapter and the LLM’s parameters

($\theta_{adapter}$ and θ_{llm}^{LoRA}). This stage is trained on a self-acquired multi-modal instruction dataset, which includes complex, scenario-specific tasks like Visual Question Answering (VQA). Given a visual input I_d and a question Q_d from the disaster domain, the model is trained to generate the correct answer A_d . The optimization objective is to minimize the standard language modeling (cross-entropy) loss:

$$\mathcal{L}_{stage2} = - \sum_{i=1}^{|A_d|} \log P(A_{d,i}|I_d, Q_d, A_{d,<i}; \theta_{adapter}, \theta_{llm}^{LoRA}) \quad (2)$$

This training transforms the VLM from a passive image descriptor into an interactive perceptual module, capable of responding to and executing vision-grounded commands.

- Stage 3 - Handling high resolution imagery: To process detailed UAV imagery effectively, we employ a dynamic local feature perception strategy with a hybrid visual encoding. It ensures the model to capture fine-grained local details while retaining crucial global contextual information during feature extraction.

During inference, this optimized VLM, denoted as $f_{vlm}(\cdot)$, functions as a dedicated perception module, processing real-time visual data I_{stream} such as video streams into structured semantic objects O_{sem} to update the *world_state*. This can be formalized as:

$$O_{sem} = f_{vlm}(I_{stream}, \text{Detect all relevant object}) \quad (3)$$

Subsequently, the core LLM, acting as the central reasoning agent, synthesizes this structured visual information with the broader context to form a coherent and actionable understanding of the operator's intention. Guided by a structured prompt template, the LLM performs high-level semantic fusion, synthesizing all information to produce a standardized "Intention XML" representation. This XML schema is meticulously engineered to serve as a direct interface for the subsequent task planning stage. A typical output includes the following key fields: $<task_type>$ defines the core action of the task, $<target>$ describes the specific object of the task (i.e., *OBJECT_ID*, *COORDINATES*) and unique identifier, $<priority>$ is used for decision-making in case of resource conflicts, $<constraints>$ contains conditions or preferences that affect tactical choices (e.g., *use_thermal_imaging*), and $<spatial_context>$ defines the geospatial scope of the task. This structured output provides a solid foundation for subsequent automated planning.

3.2 Swarm Task Planning via In-Context Learning

Upon accurately understanding the task objective through the intent grounding procedure, the this module is then responsible for decomposing the structured Intent XML into a multi-agent and parallelizable "Solution Package". A key challenge here is enabling a general-purpose LLM to execute the complex yet domain-specific reasoning this requires. Instead of the conventional path of fine-tuning on large-scale expert data, we employ a flexible In-Context Learning (ICL) strategy. The static knowledge base contains the necessary domain contextual and operational constraints, including UAV performance parameters, standard SAR tactics, and API specifications. A key advantage of this architecture is its flexibility, where the knowledge base can be modified and expanded without requiring model retraining. To structure the reasoning process, the CoT component guides the LLM through a sequential path via "Analyze \Rightarrow Retrieve \Rightarrow Assign \Rightarrow Sequence \Rightarrow Generate Code". We further reinforce this by adopting a "Code-as-CoT" paradigm, where a demonstration in the prompt conditions the model to express its final plan as executable code. This method enforces logical consistency and improves the reliability of the generated plans.

This entire inference process shapes from constructing the prompt to generating the final Solution Package is described in Algorithm 1 (see Appendix). The Solution Package integrates a natural language summary, the structured thought process, and the auditable machine instructions, providing a transparent basis for subsequent human-in-the-loop verification.

3.3 Closed-Loop Verification and Execution

This final module embeds a closed-loop, Human-in-the-Loop (HIL) verification process to ensure the LLM-generated mission is safe and reliable. This critical step serves a dual purpose, as it acts as a safeguard against logical or factual errors in the LLM's reasoning and then incorporates essential

human judgment to align the static plan with the unpredictable, dynamic conditions of the real-world operational environment. This module implements a “propose-and-confirm” interaction model, prioritizing radical transparency to facilitate informed human oversight. Upon receiving the Solution Package, the interface presents to the operator with a three-fold view, including 1) a concise natural language summary for immediate comprehension underlying the on-site environment, 2) the complete CoT rationale, available on-demand for traceability, and 3) the auditable, low-level, executable code to be implemented. By providing these insights, the system redefines the operator’s role, transforming them into a real-time monitoring and decision-making authority capable of validating the plan at both a strategic and an implementation level.

When the operator rejects a plan based on their domain expertise or real-time situational awareness unavailable to the model (e.g., a sudden gust of wind not present in the static knowledge base, or an inefficient scan pattern chosen by the model for the current terrain), their corrective feedback is treated as a high-priority constraint. This feedback triggers a re-planning cycle, prompting the Manager Agent to generate a new solution that adheres to the revised constraints. The framework proceeds to the execution stage only upon explicit operator confirmation. At this point, the executable machine instructions from the Solution Package are passed directly to a secure execution environment (e.g., the simulator or a real UAV’s API endpoint). This ensures a deterministic and verifiable transition from the audited code to physical actuation, thereby completely mitigating the risk of LLM-introduced errors during execution.

4 Experiments and Results

We evaluate the proposed LLM-Cognitive Reasoning Framework through a series of rigorously designed experiments based on a complex disaster response mission. The evaluation quantitatively compares the framework against baseline methods across three perspectives: mission success rate and planning quality, operator cognitive workload, and robustness to dynamic uncertainties.

4.1 Experimental Setup

Simulation Environment & Scene Elements & UAV Swarm Configuration. With the ethics approval obtained from our institution, this study recruited 10 UAV operators with varying levels of experience, specifically with 3 experts, 4 intermediate, and 3 novices, for task implementation and NASA-TLX self-reporting. All experiments were conducted under the same computational environment, which is AirSim [44], a high-fidelity simulator built on Unreal Engine 4. To ensure experimental diversity and reproducibility, we developed a unified, parameterized disaster scene generator that randomly creates 10 distinct site scenarios within a $2\text{km} \times 2\text{km}$ area. Each participant attempted all 4 comparative methods across all 10 scenarios, yielding a total of 400 experimental trials. The core elements and their parameter distributions for each scene are detailed in Table 1.

Table 1: Main Parameters for Randomized Disaster Scene Generation.

Element	Quantity	Scale / Intensity Parameter	Constraint / Note
Base Station	1	Coordinate: $[0, 0, 0]$	The origin for the swarm and the communication anchor for the Relay UAV.
Disaster Zone	1	Radius: 500 m Centered at a random location >600 m from Base Station	All other elements (Obstacles, Survivors) are procedurally spawned within this zone.
Obstacles	5–10	Type: Cube, Cylinder, Wall Height: 10 m to 45 m	Static obstacles. Any collision results in immediate mission failure. The model must learn their positions via mapping.
Survivors	1–5	Point source Thermal Signature: 0.8 - 1.0	Static heat sources. Primary targets for the Searcher UAV.
Wind Zones	0–2	Spherical volume Radius: 50 m to 100 m Vector: 10 m s^{-1} to 15 m s^{-1}	Dynamic Event: Appears after $t > 60$ s at a random location. Entry into this zone results in mission failure.

The UAV swarm setting comprises three heterogeneous AirSim quadrotors. The key static properties of each agent, including functional designation, sensor suite, and operational parameters, are provided to the LLM and detailed in Table 2.

Table 2: UAV Swarm Configuration Parameters.

ID	Max. Speed	Role / Sensor Payload	Primary Duty
UAV-1	10 m s^{-1}	Inspector (<i>ImageType.Scene</i>)	Performs global mapping at high altitudes ($h \in [50, 100] \text{ m}$) to provide obstacle data for the swarm.
UAV-2	10 m s^{-1}	Searcher (<i>ImageType.Infrared</i>)	Conducts low-altitude search ($h \in [10, 30] \text{ m}$) to detect and localize survivor heat signatures.
UAV-3	10 m s^{-1}	Relay (Communications Package)	Maintains a continuous communication link between the agents and the base station.

Mission Setup & Baseline Comparisons. We designed a unified, multi-objective disaster response mission, which requires all evaluated methods to generate a single, directly executable Python script. This script must autonomously coordinate three parallel objectives, including 1) *Mapping*: UAV-1 surveys the disaster zone to map obstacle locations, 2) *Searching*: UAV-2 performs low-altitude infrared scanning to locate survivors, 3) *Relying*: UAV-3 maintains a continuous communication link. The mission is governed by the following rigorous constraints, where any one of the following violation results in immediate failure. Collision avoidance, that is, no UAV collides with obstacles, UAV-2 must maintain an altitude of $h > 50 \text{ m}$ while in unmapped areas. Communication Link requires the distance between UAV-1 -2 and UAV-3 must not exceed 400m , and the distance between UAV-3 and the base station must be $\leq 1000 \text{ m}$. Dynamic Hazard Evasion suggests that no UAV allows to enter an active wind zone. All methods receive identical inputs: the initial site state (disaster zone location and UAV initial states) and complete API documentation. Specifically, we developed four configurations for experimental evaluation:

- **B1 (Manual):** Human operators manually write, debug, and execute mission code using Python and AirSim API.
- **B2 (LLM-Direct):** Base Qwen-14B-Chat model without the proposed curated Static Domain Knowledge Base or Chain-of-Thought examples.
- **B3 (Ours w/o Feedback):** Our system with ICL yet without human-in-the-loop feedback and executes the first generated plan without operator confirmation.
- **Ours (Full, LLM-CRF):** The proposed complete LLM-CRF with ICL and human feedback loop.

Evaluation Metrics. Since this work is the first of its kind to adopt an LLM in disaster SAR tasks, there is no established benchmarks for comparison. With the objective to fairly evaluate the proposed LLM-CRF, this study establishes quantitative metrics and compares its performance against baseline approaches across three dimensions, including mission success rate, task quality, and operational efficiency (i.e., operator’s cognitive workload).

Mission Success Rate (MSR): For each trial j , $MSR_j = 1$ if all objectives are met without constraint violations, else $MSR_j = 0$. Overall success rate is $MSR = \frac{1}{N} \sum_{j=1}^N MSR_j \times 100\%$.

Task Quality (TQ): For all trials (including partial completions in failures), we measure Search Coverage (Cov_{search}) as the percentage of disaster area scanned by UAV-2’s sensor footprint ($Cov_{search} = A_{covered}/A_{total} \times 100\%$), and Survivors Found ($Rate_{found}$) as the percentage of survivors correctly located ($Rate_{found} = N_{found}/N_{total} \times 100\%$).

Efficiency & Operator Load: Total Mission Time (TMT) is recorded in seconds from task start to completion (or failure). Cognitive workload is measured through the NASA Task Load Index (NASA-TLX) [45], where operators rate their task experience on six dimensions (i.e., Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, Frustration) using a 21-point scale (rated from 0–100, the lower score indicates the less cognitive load the operator has), then the complete pairwise comparisons to weight these dimensions. The final weighted score is:

$$Load_{TLX} = \frac{1}{15} \sum_i R_i \cdot w_i, \quad i \in \{\text{MD, PD, TD, Perf, Eff, Frus}\} \quad (4)$$

4.2 Experimental Results Analysis

We conducted experiments with 10 participants, each of them attempted all 4 baselines across 10 randomly generated disaster scenarios, yielding a total of 400 experimental trials. The main results are summarized in Table 3.

Table 3: Averaged Experimental Results on the Unified Mission.

Metric	B1 (Manual)	B2 (LLM-Direct)	B3 (Ours w/o Feedback)	Ours (Full)
Mission Success Rate (%)	87.0	11.0	62.0	94.0
Search Coverage (%)	94.8 ± 4.2	71.3 ± 19.8	92.3 ± 4.8	96.2 ± 2.8
Survivors Found (%)	93.1 ± 3.9	68.5 ± 21.3	79.8 ± 14.6	94.8 ± 3.1
Total Mission Time (s)	1295 ± 418	393 ± 287	387 ± 42	463 ± 51
NASA-TLX Score (%)	71.2 ± 9.3	68.5 ± 13.7	42.8 ± 8.1	28.3 ± 6.2

Overall Performance Analysis. As shown in Table 3, the proposed LLM-CRF system (Ours) demonstrates substantial superiority across all core metrics. It achieved a 94.0% Mission Success Rate, highlighting its robustness in complex, long-horizon planning under multiple constraints. In contrast, B2 (LLM-Direct), which lacks domain knowledge and structured reasoning, failed in most cases, of only 11.0% success rate obtained. This might be due to the fact that the generated code from this approach generally contains logical errors or fails to address implicit dependencies (e.g., mapping before low-altitude flight).

Regarding task quality among successful runs, the proposed LLM-CRF slightly outperformed the B1 (Manual), achieving a higher Survivors Found Rate (94.8% vs. 93.1%) with significantly lower variance (3.1% vs. 3.9%). More critically, the LLM-CRF demonstrated superior Search Coverage (96.2% vs. 94.8%), indicating more systematic and exhaustive search patterns. Analysis of failed cases revealed that human operators, though capable, were susceptible to planning fatigue, which led to suboptimal scan paths with coverage gaps.

In terms of efficiency and cognitive workload, the LLM-CRF system reduced the average mission time by 64.2% compared to manual coding (463s vs. 1295s) and lowered the NASA-TLX score by 42.9% (28.3 vs. 71.2), effectively shifting operators from a high to a low cognitive load condition. Although it significantly reduces mental workload, the proposed LLM-CRF design did not exclude the operator personnel from the operational loop or their decision-making role. Instead, it underscores the human’s critical responsibilities in monitoring and inference, which significantly enhances its potential for future deployment.

Task Complexity and the Critical Role of Human Feedback. A particularly revealing comparison exists between the full LLM-CRF system and the ablated version of B3 (Ours without Feedback). Specifically, B3 achieved the fastest execution time of 387 seconds, and demonstrated strong performance on simple subtasks. For instance, its Search Coverage of 92.3% was competitive with 96.2% of the full version and the 94.8% of B1. However, its performance deteriorated significantly on complex and safety critical tasks. The Survivors Found rate for B3 fell to 79.8%, with a high variance of $\pm 14.6\%$, and its overall Mission Success Rate dropped sharply to 62.0%. This result constitutes a 32% absolute increase in the failure rate when compared to the full system, which achieved a 94.0% success rate. This performance gap highlights a critical insight, that is, *the LLM-CRF system has already achieved human competitive competence on well-defined, static subtasks, yet human oversight remains indispensable for handling dynamic uncertainties*. Analysis of B3 failures revealed that 38% of missions failed due to:

- Collision with dynamically emerging wind zones (19% of trials)
- Suboptimal relay station placement causing communication loss (12%)
- Edge-case path-finding errors in complex obstacle fields (7%)

In contrast, when human operators reviewed the initial plans generated by the LLM-CRF and provided lightweight corrective feedback, the system successfully re-planned in 97% of flagged cases. For example, an operator might follow the instruction, “Avoid the northwest quadrant” upon receiving a weather alert. This result demonstrates that human-in-the-loop verification acts not merely as a safety net, but as a strategic necessity for bridging the gap between static reasoning and dynamic

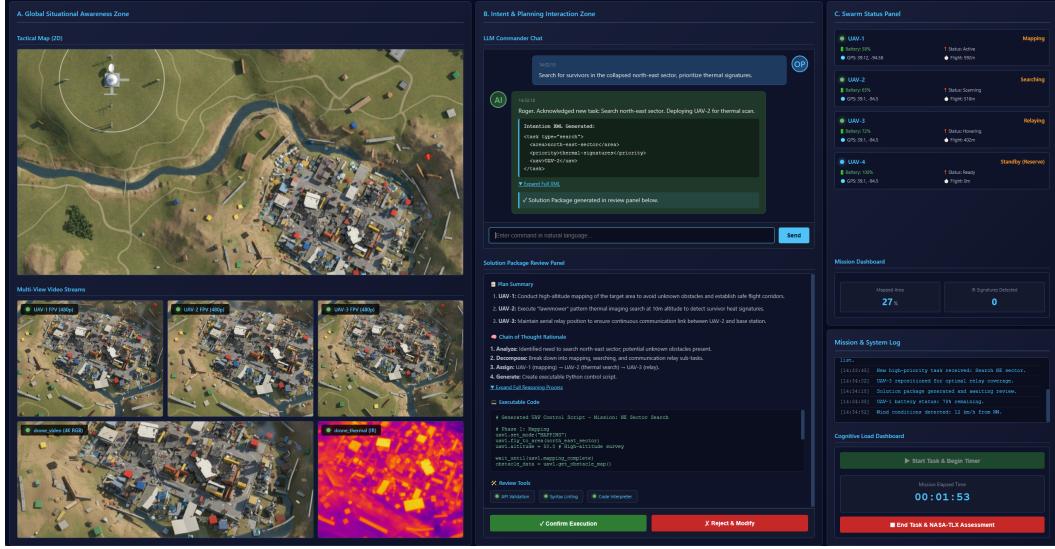


Figure 3: **The proposed LLM-CRF interface demonstration.** Left: Real-time UAV swarm site scenario (video stream and thermal feedback); Middle: LLM-CRF dialogue with generated plans; Right: Dashboard for UAV and task parameters.

reality. This collaborative approach ensured a 51.6% relative improvement in the success rate, raising it from 62.0% to 94.0%, while introducing only a minimal cognitive burden, as reflected in the NASA-TLX score which increased from 28.3 to 42.8. Figure 3 demonstrates the interface for the proposed LLM-CRF as it generates task plans and operational commands.

5 Conclusion

This paper introduces the LLM-CRF system that enables an LLM to function as an autonomous mission commander for UAV swarms under complex disaster response scenarios. The core of the proposed approach is a structured reasoning process that synergizes perceptual alignment, in-context learning, and closed-loop verification to transform a general-purpose LLM into a reliable planner, effectively bridging the gap between high-level reasoning and safe, embodied, real-case execution. A central contribution of this work is its human-on-the-loop paradigm, which redefines the operator’s role from manual coder to strategic supervisor. The framework grounds NLP commands in a real-time model, autonomously decomposes them into parallel sub-tasks (e.g., mapping, search, relay), and generates a transparent CoT rationale alongside the final executable code. This design ensures operational safety and logical soundness by mandating human validation before any action is taken.

Extensive experimental results provide robust quantitative validation of the proposed approach. Specifically, the LLM-CRF achieved a 94.0% Mission Success Rate under hard safety constraints, with a Search Coverage of 96.2% and a Survivors Found Rate of 94.8%, demonstrating its competence in generating high-quality and safe plans. Crucially, this performance was also maintained with the operator’s cognitive load (NASA-TLX) of 28.3%, confirming the framework’s success in alleviating the mental burden of complex swarm management.

While this work provides a promising foundation for human-machine teaming in critical missions, its performance is contingent on high-quality sensor data, which yields a limitation of the current evaluation. Future degradation from significant sensor noise or failures remains a key challenge. Addressing this by integrating predictive environmental models and robust dialogue-based re-planning constitutes a vital direction for future work, essential for transitioning from simulation to real-world deployment. Through fusing LLM-based strategic reasoning with human oversight, this framework provides a prior study toward deploying autonomous and reliable robotic systems.

References

- [1] S. P. H. Boroujeni, A. Razi, S. Khoshdel, F. Afghah, J. L. Coen, L. O'Neill, P. Fule, A. Watts, N.-M. T. Kokolakis, and K. G. Vamvoudakis, "A comprehensive survey of research towards ai-enabled unmanned aerial systems in pre-, active-, and post-wildfire management," *Information Fusion*, vol. 108, p. 102369, 2024.
- [2] M. T. Sadrabadi, J. Peiró, M. S. Innocente, and G. Rein, "Conceptual design of a wildfire emergency response system empowered by swarms of unmanned aerial vehicles," *International Journal of Disaster Risk Reduction*, p. 105493, 2025.
- [3] Y. Xu, Z. Jian, J. Zha, and X. Chen, "Emergency networking using uavs: A reinforcement learning approach with large language model," in *2024 23rd ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2024, pp. 281–282.
- [4] C. Chen, K. Bin, T. Hu, J. Qi, X. Liu, T. Liu, Z. Liu, Y. Liu, and P. Zhong, "Fusion meets diverse conditions: A high-diversity benchmark and baseline for uav-based multimodal object detection with condition cues," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 27 958–27 967.
- [5] C. Zhang, G. Huang, L. Liu, S. Huang, Y. Yang, X. Wan, S. Ge, and D. Tao, "Webuav-3m: A benchmark for unveiling the power of million-scale deep uav tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 7, pp. 9186–9205, 2022.
- [6] J. Liu, J. Cui, M. Ye, X. Zhu, and S. Tang, "Shooting condition insensitive unmanned aerial vehicle object detection," *Expert Systems with Applications*, vol. 246, p. 123221, 2024.
- [7] J. Zhong, J. Zhu, J. Huyan, T. Ma, and W. Zhang, "Multi-scale feature fusion network for pixel-level pavement distress detection," *Automation in Construction*, vol. 141, p. 104436, 2022.
- [8] A. L. B. Vieira e Silva, H. de Castro Felix, F. P. M. Simões, V. Teichrieb, M. dos Santos, H. Santiago, V. Sgotti, and H. Lott Neto, "Insplad: A dataset and benchmark for power line asset inspection in uav images," *International journal of remote sensing*, vol. 44, no. 23, pp. 7294–7320, 2023.
- [9] S. Wandelt, S. Wang, C. Zheng, and X. Sun, "Aerial: A meta review and discussion of challenges toward unmanned aerial vehicle operations in logistics, mobility, and monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 7, pp. 6276–6289, 2023.
- [10] T. Deng, Z. Huo, L. Zhang, Z. Dong, L. Niu, X. Kang, and X. Huang, "A vr-based bci interactive system for uav swarm control," *Biomedical Signal Processing and Control*, vol. 85, p. 104944, 2023.
- [11] Q. Peng, H. Wu, and R. Xue, "Review of dynamic task allocation methods for uav swarms oriented to ground targets," *Complex System Modeling and Simulation*, vol. 1, no. 3, pp. 163–175, 2021.
- [12] Y. Bu, Y. Yan, and Y. Yang, "Advancement challenges in uav swarm formation control: A comprehensive review," *Drones*, vol. 8, no. 7, p. 320, 2024.
- [13] M. D. Phung and Q. P. Ha, "Safety-enhanced uav path planning with spherical vector-based particle swarm optimization," *Applied Soft Computing*, vol. 107, p. 107376, 2021.
- [14] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [15] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.
- [16] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM transactions on intelligent systems and technology*, vol. 15, no. 3, pp. 1–45, 2024.
- [17] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.
- [18] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.
- [19] G. Feng, B. Zhang, Y. Gu, H. Ye, D. He, and L. Wang, "Towards revealing the mystery behind chain of thought: a theoretical perspective," *Advances in Neural Information Processing Systems*, vol. 36, pp. 70 757–70 798, 2023.
- [20] H. Qiu, J. Li, J. Gan, S. Zheng, and L. Yan, "Dronegpt: Zero-shot video question answering for drones," in *Proceedings of the International Conference on Computer Vision and Deep Learning*, 2024, pp. 1–6.
- [21] T. Li, J. Liu, W. Zhang, Y. Ni, W. Wang, and Z. Li, "Uav-human: A large benchmark for human behavior understanding with unmanned aerial vehicles," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 266–16 275.

- [22] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, “Human action recognition from various data modalities: A review,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 3, pp. 3200–3225, 2022.
- [23] R. Ribeiro, J. Ramos, D. Safadinho, A. Reis, C. Rabadão, J. Barroso, and A. Pereira, “Web ar solution for uav pilot training and usability testing,” *Sensors*, vol. 21, no. 4, p. 1456, 2021.
- [24] Q. Ouyang, Z. Wu, Y. Cong, and Z. Wang, “Formation control of unmanned aerial vehicle swarms: A comprehensive review,” *Asian Journal of Control*, vol. 25, no. 1, pp. 570–593, 2023.
- [25] S. Han, C. Fan, X. Li, X. Luo, and Z. Liu, “A modified genetic algorithm for task assignment of heterogeneous unmanned aerial vehicle system,” *Measurement and Control*, vol. 54, no. 5-6, pp. 994–1014, 2021.
- [26] F. Yan, J. Chu, J. Hu, and X. Zhu, “Cooperative task allocation with simultaneous arrival and resource constraint for multi-uav using a genetic algorithm,” *Expert Systems with Applications*, vol. 245, p. 123023, 2024.
- [27] G. M. Skaltsis, H.-S. Shin, and A. Tsourdos, “A review of task allocation methods for uavs,” *Journal of Intelligent & Robotic Systems*, vol. 109, no. 4, p. 76, 2023.
- [28] Z. Zhang, H. Liu, and G. Wu, “A dynamic task scheduling method for multiple uavs based on contract net protocol,” *Sensors*, vol. 22, no. 12, p. 4486, 2022.
- [29] M. Divband Soorati, J. Clark, J. Ghofrani, D. Tarapore, and S. D. Ramchurn, “Designing a user-centered interaction interface for human–swarm teaming,” *Drones*, vol. 5, no. 4, p. 131, 2021.
- [30] J. Zhang, Z. Yu, X. Wang, Y. Lyu, S. Mao, S. C. Periaswamy, J. Patton, and X. Wang, “Rfhui: An rfid based human–unmanned aerial vehicle interaction system in an indoor environment,” *Digital Communications and Networks*, vol. 6, no. 1, pp. 14–22, 2020.
- [31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [32] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.
- [33] J. Li, D. Li, C. Xiong, and S. Hoi, “Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation,” in *International conference on machine learning*. PMLR, 2022, pp. 12 888–12 900.
- [34] W. Dai, J. Li, D. Li, A. Tiong, J. Zhao, W. Wang, B. Li, P. N. Fung, and S. Hoi, “Instructblip: Towards general-purpose vision-language models with instruction tuning,” *Advances in neural information processing systems*, vol. 36, pp. 49 250–49 267, 2023.
- [35] S. H. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, “Chatgpt for robotics: Design principles and model abilities,” *Ieee Access*, vol. 12, pp. 55 682–55 696, 2024.
- [36] J. Zhong, M. Li, Y. Chen, Z. Wei, F. Yang, and H. Shen, “A safer vision-based autonomous planning system for quadrotor uavs with dynamic obstacle trajectory prediction and its application with llms,” in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2024, pp. 920–929.
- [37] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, “Lora: Low-rank adaptation of large language models,” *ICLR*, vol. 1, no. 2, p. 3, 2022.
- [38] H. Liu, W. Xue, Y. Chen, D. Chen, X. Zhao, K. Wang, L. Hou, R. Li, and W. Peng, “A survey on hallucination in large vision-language models,” *arXiv preprint arXiv:2402.00253*, 2024.
- [39] A. Favero, L. Zancato, M. Trager, S. Choudhary, P. Perera, A. Achille, A. Swaminathan, and S. Soatto, “Multi-modal hallucination control by visual information grounding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 303–14 312.
- [40] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang *et al.*, “Qwen technical report,” *arXiv preprint arXiv:2309.16609*, 2023.
- [41] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee, “Llavanext: Improved reasoning, ocr, and world knowledge,” 2024.
- [42] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.
- [43] Z. Zhang, T. Zhao, Y. Guo, and J. Yin, “Rs5m and georsclip: A large scale vision-language dataset and a large vision-language model for remote sensing,” *IEEE Transactions on Geoscience and Remote Sensing*, 2024.
- [44] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Aerial informatics and robotics platform,” *Washington: Microsoft Research*, 2017.
- [45] S. G. Hart and L. E. Staveland, “Development of nasa-tlx (task load index): Results of empirical and theoretical research,” in *Advances in psychology*. Elsevier, 1988, vol. 52, pp. 139–183.

A Technical Appendices and Supplementary Material

Algorithm 1 Swarm Task Planning via In-Context Learning with Chain-of-Thought

Input: Intent I , World State S

Global Context: Knowledge Base KB , API Docs API , Exemplars $E = \{(e_i^{task}, e_i^{cot}, e_i^{code})\}$

```

1: function GENERATESOLUTIONPACKAGE( $I, S$ )
2:    $Context_{domain} \leftarrow \text{RetrieveKnowledge}(I, KB); e_{sim} \leftarrow \text{FindMostSimilarExemplar}(I, E)$ 
3:    $Analysis \leftarrow \text{LLM-CoT("Analyze intent", } I, S, Context_{domain}, e_{sim}^{task}, e_{sim}^{cot})$ 
4:    $Tactics \leftarrow \text{LLM-CoT("Identify tactics", } Analysis, KB.tactics, e_{sim}^{cot})$ 
5:    $T_{tree} \leftarrow \text{DecomposeAndAssign}(I, Analysis, Tactics, KB, E)$ 
6:    $Plan_{seq} \leftarrow \text{LLM-CoT("Sequence tasks", } T_{tree}, KB.constraints, e_{sim}^{cot})$ 
7:    $\Psi_{final} \leftarrow \text{GenerateExecutableCode}(T_{tree}, Plan_{seq}, API, E)$ 
8:   return Package( $\Psi_{final}, T_{tree}, Analysis, Plan_{seq}$ )
9: end function

10: function DECOMPOSEANDASSIGN( $I, Analysis, Tactics, KB, E$ )
11:    $T_{root} \leftarrow \{task : I, role : null, subtasks : []\}$ 
12:   return RecursiveDecompose( $T_{root}, Analysis, Tactics, KB, E$ )
13: end function

14: function RECURSIVEDECOMPOSE( $T_{current}, Analysis, Tactics, KB, E$ )
15:    $e_{ref} \leftarrow \text{FindMostSimilarExemplar}(T_{current}.task, E)$ 
16:    $IsAtomic \leftarrow \text{LLM-CoT("Is atomic?", } T_{current}.task, Tactics, KB.capabilities, e_{ref}^{cot})$ 
17:   if  $IsAtomic = \text{True}$  then
18:      $T_{current}.role \leftarrow \text{LLM-CoT("Assign role", } T_{current}.task, KB.uav_roles, e_{ref}^{cot})$ 
19:     return  $T_{current}$ 
20:   else
21:      $Subtasks \leftarrow \text{LLM-CoT("Decompose", } T_{current}.task, Tactics, KB.constraints, e_{ref}^{cot})$ 
22:     for each  $st \in Subtasks$  do
23:        $T_{child} \leftarrow \{task : st, role : null, subtasks : []\}$ 
24:        $T_{child} \leftarrow \text{RecursiveDecompose}(T_{child}, Analysis, Tactics, KB, E)$ 
25:        $T_{current}.subtasks.append(T_{child})$ 
26:     end for
27:     return  $T_{current}$ 
28:   end if
29: end function

30: function GENERATEEXECUTABLECODE( $T_{tree}, Plan_{seq}, API, E$ )
31:    $CodeFragments \leftarrow \{\}$ 
32:   for each leaf  $T_{leaf} \in \text{GetLeaves}(T_{tree})$  do
33:      $uav\_id \leftarrow T_{leaf}.role; e_{code} \leftarrow \text{FindCodeExemplar}(T_{leaf}.task, uav\_id, E)$ 
34:      $code_{leaf} \leftarrow \text{LLM("Generate code", } T_{leaf}.task, API[uav\_id], e_{code}^{code})$ 
35:      $CodeFragments[T_{leaf}] \leftarrow code_{leaf}$ 
36:   end for
37:    $e_{asm} \leftarrow \text{FindAssemblyExemplar}(T_{tree}, E)$ 
38:    $\Psi_{final} \leftarrow \text{LLM("Assemble script", } T_{tree}, Plan_{seq}, CodeFragments, e_{asm}^{code})$ 
39:   return  $\Psi_{final}$ 
40: end function

41: function RETRIEVEKNOWLEDGE( $I, KB$ )
42:    $keywords \leftarrow \text{ExtractKeywords}(I)$ 
43:   return  $\{KB.tactics[k], KB.constraints[k] \mid k \in keywords\}$ 
44: end function

45: function FINDMOSTSIMILAREXEMPLAR( $task, E$ )
46:    $scores \leftarrow [\text{Similarity}(task, e_i^{task}) \mid e_i \in E]$ 
47:   return  $E[\text{argmax}(scores)]$ 
48: end function

```
