Agent in the Sky: Intelligent Multi-Agent Framework for Autonomous HAPS Coordination and Real-World Event Adaptation

Dezheng Han[1], Anbang Zhang[1], Ruxiao Chen[2], Chenyuan Feng[3], Shuaishuai Guo[1]*

[1]School of Control Science and Engineering, Shandong University, Jinan 250061, China
[2]Jonhs Hopkins University, Baltimore, Maryland, USA
[3]Department of Communication Systems, EURECOM, Sophia Antipolis 06410, France
{dezhenghan, 202234946, shuaishuai_guo}@mail.sdu.edu.cn,
rchen117@jh.edu, Chenyuan.Feng@eurecom.fr

Abstract

High Altitude Platform Station (HAPS) offers significant flexibility for dynamic adaptability and efficient user coverage. However, achieving high levels of automation in HAPS systems is fraught with challenges, particularly in comprehending complex environments and processing natural language inputs essential for autonomous operations. Existing methods, such as reinforcement learning, are task-specific and lack the ability to integrate broader environmental information. To address these limitations, we propose an Autonomous Coverage Multi-Agent (ACMA) framework, which leverages Large Language Models (LLMs) to enhance coverage through intelligent coordination of HAPS. By incorporating techniques like in-context learning, fine-tuning, and tool-calling, our framework enables agents to understand and respond to environmental cues and natural language instructions effectively. Simulation results demonstrate that the ACMA system outperforms traditional methods in coordinating coverage, adeptly managing dynamic incidents and maximizing user coverage. Compared to traditional approaches, ACMA exhibits higher intelligence and autonomy, paving the way for more adaptable and efficient HAPS systems in real-world scenarios.

Introduction

High Altitude Platform Station (HAPS) is strategically positioned within the Earth's stratosphere, typically operating at altitudes ranging from approximately 20 to 50 kilometers. These advanced platforms fulfill a dual role: firstly, they act as wireless communication nexuses, bridging connectivity gaps between urban areas and remote regions; and secondly, they function as data repositories, supplying essential computational and storage capabilities for an integrated air-to-ground network (Kurt et al. 2021). Additionally, HAPS is anticipated to be instrumental in the development of smart city frameworks and the initiation of intelligent community projects (Belmekki et al. 2024). The inherently dynamic characteristics of HAPS systems present a myriad of challenges that are contingent upon the fluctuations of environmental conditions. This complexity poses difficulties in addressing these challenges with a one-size-fits-all algorithm or model. Existing research primarily focuses on specific optimization objectives based on fixed-format inputs, restricts the capacity to leverage diverse real-world information, such as natural language descriptions of future environmental states.

Fortunately, the advent of Large Language Models (LLMs) presents a promising avenue to surmount these obstacles. Particularly, the introduction of ChatGPT (Brown et al. 2020) has marked a significant advancement, with LLMs exhibiting exceptional proficiency in comprehending and processing natural language, a capability largely due to their underlying Transformer architecture (Vaswani et al. 2017). It is crucial to recognize that HAPS relies on airships or aircraft as platforms, capable of accommodating a specific payload, which in turn enables the onboarding of the computational power required for the deployment of LLMs. Despite these advancements, current research predominantly employs LLMs as collaborative assistants to tackle specific, intricate, and laborious engineering challenges. However, such an approach underestimates the potential of LLMs in more sophisticated and dynamic environments. Instead of treating LLMs as standalone tools, a promising direction lies in integrating LLMs into pertinent inference systems. Such integration positions LLMs as an intermediary bridge, facilitating effective communication between the system and the environment. Adopting such an approach could unlock the potential to develop a more intelligent and autonomous network of agents, endowed with the capacity for adaptive decision-making and the execution of tasks in a dynamic context.

Inspired by the challenges outlined, our objective is to en-

*The work is supported in part by the National Natural Science Foundation of China under Grant 62171262 and Grant 62301328; in part by Shandong Provincial Natural Science Foundation under Grant ZR2021YQ47; in part by the Taishan Young Scholar under Grant tsqn201909043. (*Corresponding author: Shuaishuai Guo*).

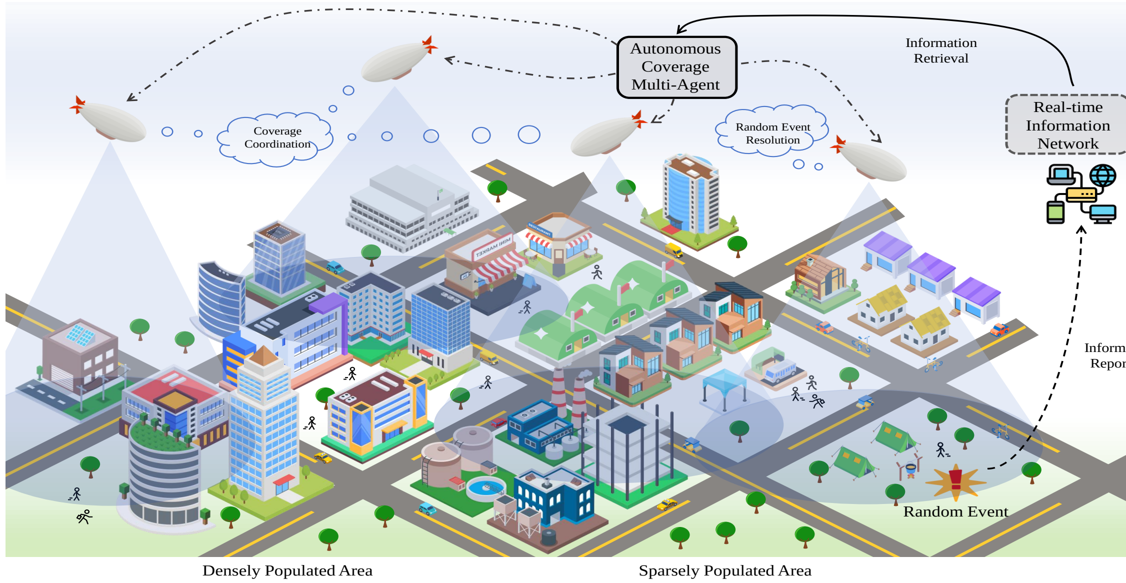

Figure 1: Example of the proposed ACMA framework within a multi-HAPS system.

hance the intelligence of autonomous agent services within the HAPS system and to unleash the perception and computing capabilities of LLMs for intelligent system applications. Thus, we propose the Autonomous Coverage Multi-Agent (ACMA) framework, designed to address the multi-HAPS coordinated communication area coverage challenge. Specifically, the research focuses on optimizing the positions of HAPS airships to ensure efficient and adaptive coverage of ground nodes. Given the highly dynamic nature of user distributions and varying network demands, the proposed framework is designed to enable HAPS airships to dynamically adjust their locations in response to changes in user density, mobility patterns, and special events. The ultimate goal is to maximize user coverage while minimizing redundancy and resource utilization, thereby ensuring seamless connectivity in complex and evolving environments. The ACMA framework consists of multiple generative agents that leverage large foundation models, enabling them to gather and analyze environmental data to make decisions aimed at maximizing user coverage. By incorporating advanced techniques such as in-context learning, fine-tuning, and tool-calling, we strive to enhance the efficacy of LLMs in tackling specific downstream tasks. The simulation results indicate that the ACMA framework excels in executing coverage maximization tasks. Furthermore, it is capable of planning and decision-making based on natural language descriptions of anticipated environments, akin to the capabilities of autonomous agents. For instance, upon receiving a natural language input such as "On March 3, 2025, at 6:00 AM, there will be a gathering at location [50, 500], ending at 3:00 PM on the same day," the system can autonomously direct a specific HAPS to the designated location in advance to provide coverage. Post-event, it seamlessly reverts to its standard coverage maximization operations. This capability signifies that when the system is equipped to receive and act upon natural language directives from the network, it can devise more intelligent strategies that align more closely with real-world requirements. As shown in Figure 1, the proposed system not only optimizes coverage but also adapts to dynamic events, showcasing its versatility and responsiveness in intelligent system applications.

Related Work

Research on the Autonomy of HAPS

Several studies on HAPS autonomy have explored reinforcement learning(RL) as a control strategy, focusing on its ability to optimize coverage by learning policies through interaction with the environment (Anicho et al. 2019). However, RL often requires extensive training time and significant computational resources, making it less practical for dynamic and real-time applications in HAPS systems. Swarm intelligence(SI) has also been investigated as a method for autonomous coordination in HAPS, leveraging decentralized algorithms inspired by collective behaviors in nature (Anicho et al. 2019). While effective in certain scenarios, SI approaches may struggle with scalability and precision, particularly in handling complex user coverage and node failure challenges in multi-HAPS networks.

Research on LLM-based agents

In the realm of LLM applications, researchers have leveraged their text comprehension capabilities to accomplish various tasks, such as automating the generation of technical reports (Wang et al. 2024b), building related semantic communication systems (Guo et al. 2023), (Guo et al. 2024), and addressing complex problem modeling in satellite communications (Zhang et al. 2024) and base station siting (Wang et al. 2024a). LLMs are also adept at simulating human behaviors and social interactions (Park et al. 2023), and facilitating collaborative tasks through dialogues (Qian et al. 2024). Some advanced agents can even under-

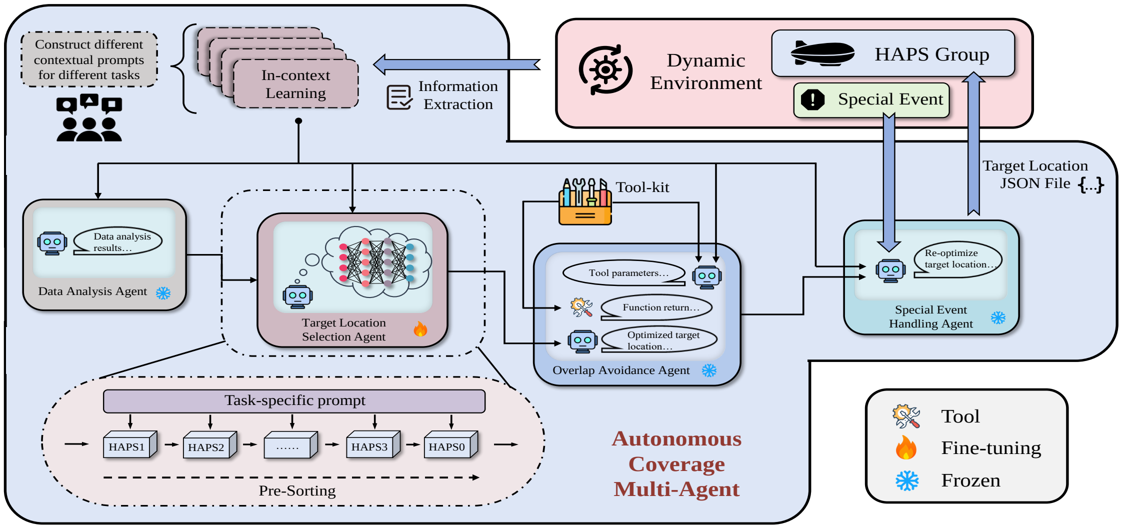

Figure 2: An illustration of the internal architecture of our proposed ACMA framework, which incorporates LLM-enhanced techniques such as fine-tuning, in-context learning, and tool-calling.

stand user needs and operate computers similarly to humans (Anthropic 2024). Our goal is to harness these capabilities to develop intelligent cooperative agents that enhance autonomous services in systems, setting our work apart from typical LLM-based approaches.

Proposed Methods

The proposed ACMA framework, as shown in Figure 2, incorporates a Multi-Agent workflow to decompose the overall coverage task into four subtasks. To enhance the ability of different agents to handle their respective tasks, we integrate in-context learning, fine-tuning, and tool-calling techniques. Constructed on LangChain (LangChain 2024), our architecture leverages OpenAI's GPT-4o-mini (OpenAI 2024) as the foundation model that powers the agents. It is noteworthy that our framework is a general-purpose architecture and is not restricted to LangChain or GPT-4o-mini. The system ingests environmental information, including metrics like the number of users covered and the positions of HAPS, as its input. This data is subsequently processed through the multi-agent workflow, culminating in the generation of a JSON-formatted file that outlines the target locations. The location data encapsulated within this JSON file is then utilized to control the HAPS system.

Precisely, we integrate pertinent environmental data into crafted prompts, which act as a unified input for the LLM. Tailoring prompts for various tasks, we employ in-context learning to ensure the LLM to generate outputs that align more closely with our requirements. To handle the four distinct subtasks, we designed intelligent agents with the following four functionalities:

- **Data Analysis Agent:** This agent is tasked with extracting "high-quality locations" from both historical and current environmental data. "High-quality locations" are defined as coordinates that exhibit high user density, significant coverage demands, or strategic importance for communication efficiency. These locations are identified and offered as reference points for subsequent agents to optimize their decision-making processes.
- **Target Location Selection Agent:** This agent is responsible for making the primary decisions. It synthesizes judgments based on the current environmental data and the 'high-quality locations' provided by the Data Analysis Agent. In a multi-HAPS system, the decision-making process is structured in a sequential manner. This sequential selection process allows for informed and logically coordinated outcomes, where decisions of one agent influence those of the next.
- **Overlap Avoidance Agent:** This agent refines the target locations determined by the Target Location Selection Agent. It detects and optimizes locations to mitigate significant coverage overlaps, thereby enhancing resource efficiency and minimizing redundancy.
- **Special Event Handling Agent:** This agent can interpret natural language descriptions of unexpected events and adapt the decision-making process, ensuring the system's agility and responsiveness to real-time changes.

The output provided by our designed agents includes a JSON-formatted file, which facilitates direct extraction and standalone use. JSON (JavaScript Object Notation) is a lightweight data-interchange format widely used for its simplicity and readability. Its structure is based on key-value pairs and supports nested data types, making it ideal for representing hierarchical information. Given the inherent variability in the outputs of LLMs, relying solely on in-context learning templates may not ensure consistent formatting. Therefore, we have implemented a fine-tuning strategy for the target location selection agent by using a small dataset of

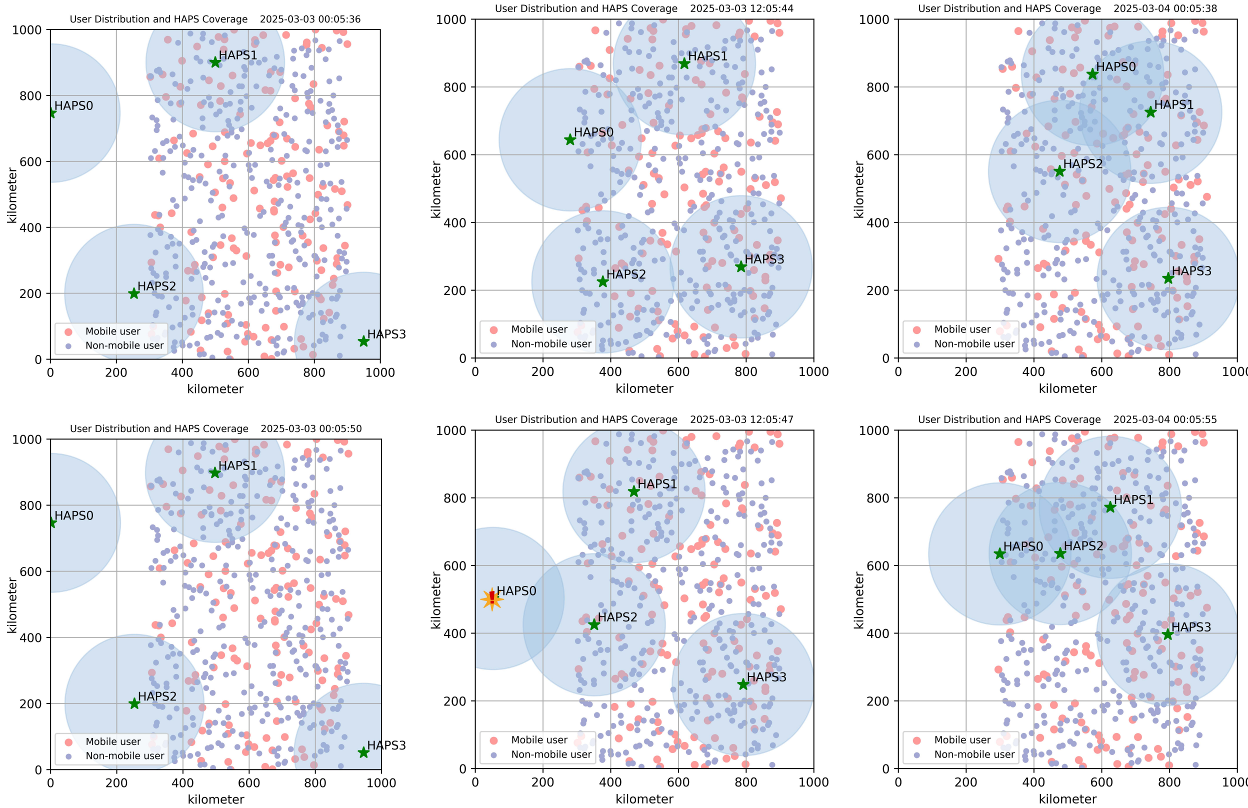

Figure 3: Visualization of user distribution and HAPS coverage. The first row shows the execution strategy under normal conditions without special events. The second row shows the execution strategy when the system encounters a special event where the red exclamation mark indicates a burst user gathering event.

curated exemplary dialogues. This fine-tuning process bolstered the proficiency of our designed agents in producing standardized JSON, which are pivotal for the target location selection subtask.

To address the issue of coverage overlap, precise mathematical computations are essential, including the determination of distances between various locations. Given that LLMs are primarily designed for probabilistic language prediction rather than precise numerical computations, they may encounter inaccuracies, especially in operations involving floating-point arithmetic. To counteract these potential inaccuracies, we have developed a straightforward tool-calling function accessible to our agent. To ensure the accuracy and reliability of our system's spatial analysis, this function is tasked with computing the pairwise distances between HAPS and generates a text prompt that encapsulates the results of these calculations in the following format:

```
Distance between HAPS0 and HAPS1: 522.02 km
Distance between HAPS0 and HAPS2: 604.15 km
...
```

The agent interprets the function by analyzing its description, generates the required input parameters, and executes the function accordingly. The output text returned by the function is then fed back into the agent as new input to produce the final optimized result. To ensurethat the data is both machine-readable and actionably informative, the system's output, encompassing a JSON file, is meticulously structured in the following format:

```
"2025-03-03 00:00:00": {
    "HAPS0": {
        "target_position": ,
        "reasoning": {}
    },
    ...
}
```

This JSON file records the start time of the decision-making process, the target locations assigned to each HAPS, and the corresponding reasoning. By extracting the `target_position` coordinates for each HAPS from this file, we are able to exert precise control over the positioning within the HAPS system, ensuring that the operational directives are executed with accuracy and efficiency.

Simulation & Results

To assess the efficacy of the proposed ACMA framework, we established a tailored simulation environment for interfacing with our system. In the simulation, it is assumed that there are 4 HAPS airships and a population of 500 users, each employing a variety of mobility patterns. To focus on the evaluation of the capabilities of LLM-based agents in

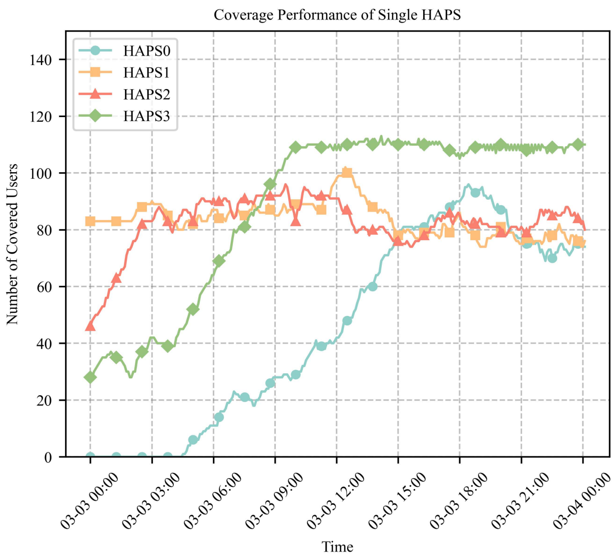

Figure 4: Coverage performance for each individual HAPS Airship.

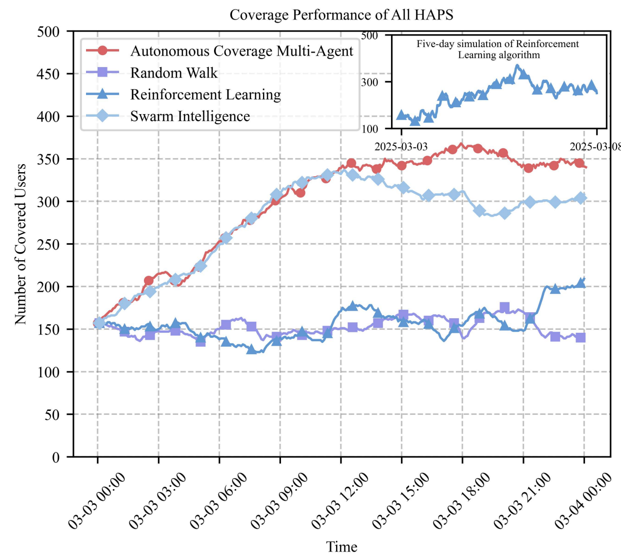

Figure 5: Coverage performance comparison under different control strategies.

understanding and executing tasks, we intentionally streamlined the physical representation of the HAPS airships, prioritizing the agents' cognitive and operational assessments over detailed mechanical simulations. The deployment location of the LLM is a key consideration in our framework. Our ultimate goal is to deploy LLMs directly onboard HAPS airships to enable autonomous decision-making with minimal dependence on ground infrastructure. However, due to current hardware constraints, our simulation experiments rely on accessing LLMs through internet-based requests to ground servers.

Considering factors such as communication quality and flight altitude, we defined the coverage radius of the HAPS airships (Guan, Yuan, and Guo 2009) as approximately 210 km, based on a flight altitude of 22 km and a minimum communication elevation angle of 5°, and adopted the concept of "timestamps" to simulate a 24-hour operational cycle. At the start of each hour within this cycle, the ACMA system determines the "target positions" for the HAPS airships based on their current locations and the extent of user coverage. These coordinates are then utilized by the simulation environment to control the simulated movement of the HAPS airships, while continuously refreshing the environmental data to reflect real-time changes throughout the simulation. This approach allows for a dynamic and responsive assessment of the system's performance in managing the airships' coverage areas.

To visually represent the environmental dynamics, we generated three coverage maps over a 24-hour period for two simulation scenarios, as depicted in Figure 3. The first row of images illustrates the control strategies of the ACMA framework under normal conditions, with no special events occurring. Under normal conditions, the control strategy aims to maximize user coverage by dynamically adjusting the positions of the HAPS airships. The system leverages the Data Analysis Agent to extract high-density user regions and the Target Location Selection Agent to assign optimal positions to each HAPS. Overlap Avoidance Agents ensure minimal redundancy, resulting in efficient resource utilization. The second row shows the execution strategy when the system encounters a special event indicated by the following natural language description:

```
On March 3, 2025, at 6:00 AM, there will be a
gathering at location [50, 500], ending at 3:00 PM on
the same day.
```

It has been observed that the system adeptly detects areas with a high user density and navigates the HAPS airships to target these zones effectively. Additionally, upon the occurrence of a gathering event at the coordinates [50, 500], our system is proficient in dispatching an HAPS airship to ensure coverage for the specified location. Post-event, the system seamlessly reallocates the HAPS airship to return to its regular coverage operations. This capability underscores the system's proficiency in comprehending natural language descriptions of events and devising and executing fitting responses, highlighting its adaptability and effectiveness in dynamic operational scenarios.

Figure 4 illustrates the coverage performance of individual HAPS airships during the simulation process. By analyzing the trends in the number of covered users, it is evident that the ACMA framework demonstrates robust dynamic scheduling capabilities. Initially, some HAPS airships have relatively low user coverage; however, the ACMA framework effectively drives these HAPS to relocate to high user-density areas, significantly increasing their coverage. Furthermore, in the later stages of the simulation, the number of covered users gradually stabilizes, indicating that the ACMA framework is not only capable of efficiently responding to dynamic user distribution changes but also ensures resource allocation stability as the system reaches equilibrium. Among the four HAPS airships, the user coverage shows a certain degree of balance, reflecting the fairness and efficiency of the resource scheduling.

Figure 5 depicts the cumulative number of users covered by all HAPS airships under different control policies. In this work, our proposed solution will be compared with the following three existing strategies: random walk, reinforcement learning, and swarm intelligence algorithms (Anicho et al. 2019). It can be observed that our algorithm consistently provides the maximum user coverage, followed by the swarm intelligence algorithm, while the random walk and reinforcement learning algorithms perform the worst. It is worth mentioning that in our simulations using the reinforcement learning algorithm, despite multiple attempts, its training did not converge in the short term. This limitation may stem from the specific characteristics of the simulation environment or to the intricacies of the reward function's design. This finding underscores a constraint inherent in traditional algorithms, necessitating a high level of domain-specific expertise from engineers for their effective deployment. In contrast, the LLM-based agents in our ACMA architecture, especially those with tool-calling capabilities, can effectively execute similar tasks through natural language instructions, markedly lowering the expertise threshold required for engagement. Moreover, its natural language processing capabilities enable it to handle random events in complex real-world environments, offering a notable advantage in practical applications.

Conclusion & Discussion

This work demonstrated the effectiveness of the proposed ACMA architecture in addressing the multi-HAPS coordination task for user coverage, and it validated the capability of the LLM-based agent framework to render more intelligent judgments regarding unpredictable events in real-world scenarios. With the continuous advancements in LLM quantization technologies and improvements in edge computing capabilities, an environment and event-adaptive agent framework will emerge as a promising solution soon. Such frameworks, deployable at the edge, are poised to tackle human-machine interaction challenges within systems that necessitate a degree of autonomy. By providing solutions to real-time challenges, these frameworks will enable more intelligent services for users and contribute to building a more cohesive and intelligent network.

References

Anicho, O.; Charlesworth, P. B.; Baicher, G. S.; Nagar, A.; and Buckley, N. 2019. Comparative Study for Coordinating Multiple Unmanned HAPS for Communications Area Coverage. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 467–474.

Anthropic. 2024. Claude 3.5 Overview. https://www.anthropic.com/claude. Accessed: 2024-11-10.

Belmekki, B. E. Y.; Aljohani, A. J.; Althubaity, S. A.; Harthi, A. A.; Bean, K.; Aijaz, A.; and Alouini, M.-S. 2024. Cellular Network From the Sky: Toward People-Centered Smart Communities. *IEEE Open Journal of the Communications Society*, 5: 1916–1936.

Brown, T.; Mann, B.; Ryder, N.; and et al. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Guo, S.; Wang, Y.; Li, S.; and Saeed, N. 2023. Semantic Importance-Aware Communications Using Pre-Trained Language Models. *IEEE Communications Letters*, 27(9): 2328–2332.

Guo, S.; Wang, Y.; Ye, J.; Zhang, A.; and Xu, K. 2024. Semantic Importance-Aware Communications with Semantic Correction Using Large Language Models. arXiv:2405.16011.

Kurt, G. K.; Khoshkholgh, M. G.; Alfattani, S.; Ibrahim, A.; Darwish, T. S.; Alam, M. S.; Yanikomeroglu, H.; and Yongacoglu, A. 2021. A vision and framework for the high altitude platform station (HAPS) networks of the future. *IEEE Communications Surveys & Tutorials*, 23(2): 729–779.

LangChain. 2024. LangChain - Building applications with LLMs made simple. https://www.langchain.com/. Accessed: 2024-11-8.

OpenAI. 2024. OpenAI Platform Documentation - Overview. https://platform.openai.com/docs/overview. Accessed: 2024-11-17.

Park, J. S.; O'Brien, J.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–22. New York, NY, USA: Association for Computing Machinery.

Qian, C.; Liu, W.; Liu, H.; Chen, N.; Dang, Y.; Li, J.; Yang, C.; Chen, W.; Su, Y.; Cong, X.; Xu, J.; Li, D.; Liu, Z.; and Sun, M. 2024. ChatDev: Communicative Agents for Software Development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15174–15186. Bangkok, Thailand: Association for Computational Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, Y.; Afzal, M. M.; Li, Z.; Zhou, J.; Feng, C.; Guo, S.; and Quek, T. Q. S. 2024a. Large Language Models for Base Station Siting: Intelligent Deployment based on Prompt or Agent. arXiv:2408.03631.

Wang, Y.; Guo, Q.; Yao, W.; Zhang, H.; Zhang, X.; Wu, Z.; Zhang, M.; Dai, X.; Zhang, M.; Wen, Q.; Ye, W.; Zhang, S.; and Zhang, Y. 2024b. AutoSurvey: Large Language Models Can Automatically Write Surveys. arXiv:2406.10252.

Zhang, R.; Du, H.; Liu, Y.; Niyato, D.; Kang, J.; Xiong, Z.; Jamalipour, A.; and Kim, D. I. 2024. Generative AI Agents with Large Language Model for Satellite Networks via a Mixture of Experts Transmission. arXiv:2404.09134.