When LLM Agent Teams Fail at Topological Spatial Reasoning Under Partial Observability

Extended Abstract

Heitor Gama
Escola Politécnica - USP, Brazil
IRL Crossing
Adelaide, Australia
heitor_gama@usp.br

Jean-Philippe Diguet
CNRS
IRL Crossing
Adelaide, Australia
jean-philippe.diguet@cnrs.fr

Damith C. Ranasinghe
Adelaide University
IRL Crossing
Adelaide, Australia
damith.ranasinghe@adelaide.edu.au

ABSTRACT

Multi-agent planning and coordination remain challenging in partially observed environments. Large language models (LLMs) offer a solution by enabling text-native agents capable of planning and communicating in natural language. In this study, we examine a failure-prone ingredient for scalable coordination. In particular, we examine LLM-agents' planning capabilities over map-like topologies, commonly encountered in navigation tasks. We attempt to stress topological spatial reasoning under decentralized information to assess their limitations. Through a task formulation, we isolate and study spatial reasoning failures hindering scalable coordination and share our findings to help advance methods to improve topological spatial reasoning. Supplementary material at https://doi.org/10.5281/zenodo.18694368

KEYWORDS

LLMs; Multi-Agent Systems; Spatial Reasoning; Graph Navigation; Decentralized Coordination

ACM Reference Format:
Heitor Gama, Jean-Philippe Diguet, and Damith C. Ranasinghe. 2026. When LLM Agent Teams Fail at Topological Spatial Reasoning Under Partial Observability: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/RTFT9958

1 PROBLEM MOTIVATION

Teams of autonomous agents often need to coordinate navigation [6] and execute tasks in environments with complex connectivity [10, 12]. Multi-agent planning is a hard problem in general [2, 10]. LLM agents offer a practical mechanism for coordination through dialogue and instruction following, so recent work studies LLM-agent teams in cooperative task settings [4, 5, 15]. We are motivated to study cooperative tasks where navigation is central to task success and each agent receives only local observations to understand the limits of LLM-based agents [5, 10]. In this study, we focus on studying topological spatial reasoning of agents, critical for path planning, under partial observability [9, 13, 14].

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/RTFT9958

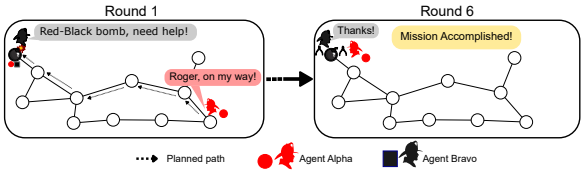

Figure 1: Overview of the LLM-based multi-agent team task in which each firefighter, an independent text-only LLM-agent, communicates with teammates to defuse a bomb.

2 TASK MODEL

To conduct the study, we specify a task model, as described next, to support the systematic analysis of map representations and memory design for LLM multi-agent teams.

Environment graph. We base the task on cooperative text benchmarks for LLM coordination [5] and include optional features for custom maps to support our study. In this scenario, we model the map as an undirected, unweighted, connected graph $G = (V, E)$ [1]. Each node represents a room and each edge represents a corridor. For a node $v \in V$, the neighbor set is $\mathcal{N}(v) \coloneqq \{u \in V \mid \{u, v\} \in E\}$.

Agents, hidden entities, and objective. We consider a team of N agents $\mathcal{I} = \{1, \ldots, N\}$ that acts in discrete rounds. The environment contains a fixed set of hidden entities tied to the objective. We use a bomb-defusal instance, adapted from prior cooperative text benchmarks [5], to instantiate the model.

Let $\mathcal{B} = \{1, \ldots, K\}$ denote bombs. Each bomb $b \in \mathcal{B}$ has a location $\mathrm{loc}(b) \in V$ and an internal phase sequence over a finite alphabet C. Agents only observe bomb information through local interaction in the same room. The team succeeds when it completes all phases for all bombs before a horizon $T_{\max}$.

State, actions, observations, & communication At round t, the latent state contains the map G, agent positions $x_t \in V^N$, and hidden progress variables for bombs. Each agent selects (i) an environment action and (ii) a chat message. We use three action types:

$$\mathcal{A}_i = \{\mathrm{move}(u) \mid u \in \mathcal{N}(x_t^i)\} \cup \{\mathrm{inspect}(b)\} \cup \{\mathrm{cut}(c, b)\},$$

where inspect reveals task-relevant hidden information for a co-located bomb and cut attempts to advance a bomb phase using an agent tool. The environment returns a local observation to the acting agent, containing the current room id, local neighbors, locally present bombs, optional probe feedback, and the public chat log $\mathcal{L}_{t-1}$. The chat log aggregates messages from earlier rounds and serves as the sole synchronization mechanism for hidden discoveries and intended plans.

Partial observability. Each agent receives local observations tied to its position. Hidden bomb states remain private until an agent visits and inspects. Team-level knowledge only arises through communication [2, 5]. The model therefore combines topological planning over G with decentralized information aggregation over $\mathcal{L}_t$.

Action interface and invalid outputs. The text interface parses agent outputs into actions. Agents receive preset rule-based feedback of their invalid actions, enabling analysis of invalid-action cascades under long-horizon prompting [11, 18].

2.1 Map-to-Text Interfaces Formulation

The task model requires an interface that renders a map to text. Let enc_{map} denote a deterministic graph-to-text encoding. The encoding choice controls prompt length and ambiguity [3, 13, 14]. In this work, we use the full-map encoding where each agent receives the entire $\text{enc}_{\text{map}}(G)$ at every turn [5].

We also use an explicit belief template that agents maintain as internal memory [5]. The template stores a structured summary of agent position, known bombs, inferred teammate locations, tools, and recent messages. Agents update the template each round using their latest observation and the chat log. Since an LLM generates updates, the belief may accumulate inconsistencies, enabling study of self-consistency under long-horizon planning [11].

2.2 Performance Evaluation.

Score: We evaluate team performance using a cumulative score that rewards successful bomb defusal. When an agent completes the final phase of a bomb, the team receives $10 \cdot \phi(b)$ points, where $\phi(b) \in \{1, 2, 3\}$ denotes the number of phases for bomb b. Under the canonical configuration with two single-phase bombs, two double-phase bombs, and one triple-phase bomb, the maximum achievable score equals 90 points. **Rounds to Completion:** This measures efficiency and is defined as the number of rounds elapsed until the episode reaches a terminal success state where all bombs are fully defused. If the team fails to defuse all bombs before the horizon $T_{\max}$, we set Rounds to Completion to $T_{\max} = 100$.

3 EMPIRICAL STUDY

We study scalable coordination by focusing on map size. This factor can be adjusted in a concrete way, allowing us to isolate its effects empirically without changing the underlying task dynamics [8, 13, 16]. We describe the settings and the experiment designed below.

Map-to-text encoding burden. Agents require a textual representation of the map to plan routes and coordinate. We provide each agent with a full-map encoding (natural-language description or canonical adjacency list) at each decision step [3, 13, 17]. Since serialization length grows with $|V| + |E|$, larger maps consume a substantial fraction of the prompt budget and force long-horizon reasoning over order-sensitive text.

Decentralization and synchronization via chat. Team knowledge aggregates through a public message log. We choose the size (in rounds) of the chat history buffer, that is, we remove the oldest message to add the newest after the buffer is full [4, 7, 15].

Memory representation under token limits. We use a structured belief template to summarize state-relevant information. The template functions as an explicit memory interface that competes with raw chat history for tokens [5, 11].

The new formulation supports controlled studies that vary one ingredient at a time while holding remaining ingredients fixed. This protocol enables attribution of coordination failures to map serialization burden; importantly it provides an empirical basis for subsequent method development to study observability constraints, memory design, and communication reliability.

Experiment. Under full-map encoding with no communication dropout, 3 agents and 5 bombs, we vary the map size.

Table 1: Scaling size. Performance degrades on large maps.

Model	#Nodes	Valid Action %	Rounds to Completion	Score
o4-mini	5	93.80%	13 ± 3.00	90 ± 0.00
	8	97.22%	14 ± 3.46	90 ± 0.00
	16	98.80%	20 ± 3.46	90 ± 0.00
	53	95.25%	79.33 ± 28.22	83.33 ± 11.55
	100	90.33%	100 ± 0	20 ± 8.17
o3-mini	5	91.24%	14 ± 1.73	90 ± 0.00
	53	86.57%	91.33 ± 12.26	63.33 ± 24.94
Llama-3.1-70B	5	62.91%	19 ± 9.29	80 ± 17.32
	53	12.79%	100 ± 0	6.67 ± 4.71

3.1 Results and Findings

Table 1 shows a strong dependence on model class. The reasoning-oriented models (o4-mini and o3-mini) remain near-optimal on small maps and only exhibit marked degradation as node count increases, with longer episodes and reduced score on the largest instances. In contrast, Llama-3.1-70B shows low Valid Action % already on the 5-node map, indicating early interface-level failures and making it a weaker choice for this task family. The gap suggests that failures at scale for o4/o3 stem more from topological planning under long contexts, motivating map encoding strategies that can recover their performance.

Our study discovers that when scaling beyond trivial, fully visible spatial layouts or *maps* to large, partially observable maps, we observe sharp drops in spatial coherence and team coordination for zero-shot agents. In these scenarios, partial observability, limited message budgets, and latency pressure reduces agent ability to maintain consistent world models while planning through natural language communication.

3.2 Future Work

The task model we describe supports systematic evaluation along axes that matter for scalable LLM coordination. **(i) scaling**, where map serialization length and topology interact with token budgets under long-horizon navigation; **(ii) interfaces**, where full-map encodings compete with region-collapsed local views under matched prompt budgets; and **(iii) memory**, where structured belief templates trade off against long chat transcripts for decision quality.

In the future, we plan to study failure modes from **communication**, such as how message dropouts shape coordination, error propagation, and redundancy in exploration as well as propose solutions to improve spatial reasoning of LLM-agents [4, 7, 15]. This sets the stage for a full empirical study in an extended work.

ACKNOWLEDGMENTS

This work was funded by the IRL CROSSING Internship Grant 2025.

REFERENCES

[1] J. Adrian Bondy and Uppaluri S. R. Murty. 2008. *Graph Theory*. Springer.
[2] Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R. McKee, Joel Z. Leibo, Kate Larson, and Thore Graepel. 2020. Open Problems in Cooperative AI. *CoRR* (2020).
[3] Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. 2024. Talk like a Graph: Encoding Graphs for Large Language Models. In *ICLR*.
[4] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. In *NeurIPS*.
[5] Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Charles Lewis, and Katia P. Sycara. 2023. Theory of Mind for Multi-Agent Collaboration via Large Language Models. In *EMNLP*.
[6] Hoa Van Nguyen, Ba-Ngu Vo, Ba-Tuong Vo, Hamid Rezatofighi, and Damith C Ranasinghe. 2024. Multi-Objective Multi-Agent Planning for Discovering and Tracking Multiple Mobile Objects. *IEEE transactions on signal processing* 72 (2024).
[7] Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *UIST*.
[8] Pengrui Quan, Brian Wang, Kang Yang, Liying Han, and Mani Srivastava. 2025. Benchmarking Spatiotemporal Reasoning in LLMs and Reasoning Models: Capabilities and Challenges. *CoRR* (2025).
[9] Md Imbesat Hassan Rizvi, Xiaodan Zhu, and Iryna Gurevych. 2024. SpaRC and SpaRP: Spatial Reasoning Characterization and Path Generation for Understanding Spatial Reasoning Capability of Large Language Models. In *ACL*.
[10] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R. Sturtevant. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artif. Intell.* (2015).
[11] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *NeurIPS*.
[12] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinícius Flores Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *AAMAS*.
[13] Jianheng Tang, Qifan Zhang, Yuhan Li, Nuo Chen, and Jia Li. 2025. GraphArena: Evaluating and Exploring Large Language Models on Graph Computation. In *ICLR*.
[14] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023. Can Language Models Solve Graph Problems in Natural Language?. In *NeurIPS*.
[15] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework. *CoRR* (2023).
[16] Hao Xu, Xiangru Jian, Xinjian Zhao, Wei Pang, Chao Zhang, Suyuchen Wang, Qixin Zhang, Joao Monteiro, Qiuzhuang Sun, and Tianshu Yu. 2025. GraphOmni: A Comprehensive and Extendable Benchmark Framework for Large Language Models on Graph-theoretic Tasks. *CoRR* (2025).
[17] Yutaro Yamada, Yihan Bao, Andrew Kyle Lampinen, Jungo Kasai, and Ilker Yildirim. 2024. Evaluating Spatial Understanding of Large Language Models. *TMLR* (2024).
[18] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *ICLR*.