On the Reliability of LLM Orchestration Under Horizontal Knowledge Partitioning

Mayar Osama, Nourhan Ehab and Mervat AbuElkheir

German University in Cairo, Egypt

Abstract

Large Language Models (LLMs) are increasingly used as coordinators in multi-agent systems (MAS), particularly in settings composed entirely of LLM-based agents; however, their ability to coordinate symbolic or hybrid agents under partial observability remains underexplored. In this paper, we investigate whether LLM-based orchestrators can reliably detect and resolve coordination deadlocks in heterogeneous MAS with horizontally partitioned knowledge. We evaluate this problem using a distributed zebra-style constraint satisfaction task involving three specialized agents with restricted reasoning capabilities: equivalence reasoning, relational reasoning, and negation reasoning. We compare fully neural (LLM-only) and neural-symbolic (LLM + Prolog) execution settings under the same orchestration framework, augmented with explicit plan generation, symbolic verification, execution monitoring, and structured planning traces. Our results show that while LLM-based coordination performs effectively in LLM-only settings, it struggles to resolve stagnation and deadlocks when coordinating symbolic agents, despite the symbolic agents maintaining high role-faithfulness and logically valid local reasoning. We further observe that successful LLM-only coordination frequently coincides with violations of role-restricted reasoning boundaries, suggesting that apparent coordination success may partially arise from cross-partition inference leakage rather than robust long-horizon planning. These findings highlight a failure mode in LLM orchestration for heterogeneous MAS and motivate evaluation metrics that consider role compliance and coordination validity in addition to task completion.

Keywords

Large Language Models, Multi-Agent Systems, NeuroSymbolic, Knowledge Reasoning

1. Introduction

Large Language Models (LLMs) are increasingly used as planners and orchestrators in multi-agent systems (MAS), where they coordinate specialized agents to solve complex tasks through iterative interaction and information exchange. Recent orchestration frameworks have demonstrated strong performance in LLM-only settings by combining centralized planning, structured communication, and adaptive task allocation. However, most existing evaluations focus on systems in which all participating agents are themselves LLMs or share compatible reasoning and communication mechanisms. Much less is understood about how LLM planners behave when coordinating heterogeneous systems composed of symbolic and non-symbolic agents with strictly partitioned knowledge and incompatible reasoning procedures.

At the same time, prior work on LLM planning has repeatedly shown that LLM-generated plans degrade under long-horizon reasoning, constraint satisfaction, and execution monitoring requirements[1, 2, 3, 4]. Common failures include invalid action sequencing, inconsistent state tracking, weak replanning behavior, and inability to systematically reason over constrained search spaces[5, 6, 7]. These limitations become particularly important in heterogeneous multi-agent coordination, where progress depends not only on local reasoning quality but also on the planner's ability to strategically query agents, integrate partial information, detect stagnation, and revise execution strategies under partial observability. Despite extensive work on LLM planning and orchestration, the specific problem of coordination stagnation under strict horizontal partitioning remains underexplored.

Joint Workshop on Statistics and Knowledge Integration for Logic, Learning, Ethical Decisions, and LLMs (SKILLED-LLMs 2026), co-located with FLoC 2026, Lisbon, Portugal

mayar.osama@guc.edu.eg (M. Osama); nourhan.ehab@guc.edu.eg (N. Ehab); mervat.abuelkheir@guc.edu.eg (M. AbuElkheir)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-07-23

CEUR-WS.org/Vol-4229/paper19.pdf

In this paper, we study LLM orchestration in a cooperative constraint-satisfaction setting where knowledge is horizontally partitioned across three specialized agents. Each agent has access only to a restricted subset of constraints: equivalence reasoning, relational reasoning, or negation-based reasoning. No individual agent can solve the task independently, and successful execution requires coordinated propagation of partial inferences across agents. We compare two execution settings under the same orchestration loop: an LLM-only system in which execution agents are implemented as role-constrained LLMs, and a hybrid neural-symbolic system in which execution agents are implemented as symbolic Prolog-style solvers.

Our experiments show a consistent behavioral difference between these settings. Symbolic-agent executions exhibit high role fidelity and locally valid reasoning, yet frequently enter coordination stagnation in which the planner repeatedly queries agents without generating meaningful global progress. In contrast, LLM-only executions often achieve greater apparent progress, but may violate role-local reasoning boundaries by producing eliminations or inferences not justified by the agent's visible constraints. We further investigate whether orchestration failures can be mitigated through neuro-symbolic interventions, including symbolic planner-step verification, structured Prolog-style prompting, and explicit plan-review mechanisms. Under these interventions, the hybrid orchestration loop achieves a fully correct solution on the benchmark task.

Accordingly, this paper investigates the following research questions:

- **RQ1**: Can LLM planners resolve coordination stagnation under strict horizontal partitioning?
- **RQ2**: Why do LLMs plan better for LLM-only systems than planning for symbolic-agent systems?
- **RQ3**: Can orchestration failures be mitigated with neuro-symbolic interventions?

The remainder of this paper is organized as follows. Section 2 reviews related work on LLM-based multi-agent planning, orchestration failures, neuro-symbolic systems, and structured reasoning approaches. Section 3 presents the experimental framework, including the horizontally partitioned constraint-satisfaction benchmark, agent partitioning strategy, orchestration loop, execution conditions, and evaluation metrics. Section 4 reports the experimental results and analyzes coordination stagnation, role/persona compliance, planner failure patterns, and prompting ablations across LLM and symbolic execution settings. Finally, Section 5 concludes the paper and discusses directions for future work.

2. Background and Related Work

Recent LLM-based multi-agent systems (LLM-MAS) increasingly rely on centralized orchestration architectures in which a planner coordinates multiple specialized execution agents through iterative delegation and shared memory [8, 1]. Prior work shows that structured orchestration improves coordination and task decomposition relative to monolithic single-agent execution. However, most existing evaluations are conducted in settings where both planners and execution agents are themselves LLM-based [1], making it difficult to isolate whether observed coordination success depends on robust orchestration itself or on the semantic flexibility of unconstrained LLM execution agents.

At the same time, recent work shows that LLM agents struggle to sustain coherent long-horizon planning, accurate state tracking, and stable coordination over extended interactions [9, 10]. Planning failures frequently emerge as repeated ineffective actions, context drift, and cascading execution inconsistencies [9]. These limitations become more severe in collaborative settings, where multi-agent systems exhibit misinformation propagation, belief-state inconsistencies, and oscillatory coordination behavior [11, 12]. Together, these findings suggest that coordination stagnation is a structurally plausible outcome of iterative LLM orchestration rather than merely an implementation artifact.

The difficulty of coordination under restricted local knowledge has long been studied in distributed constraint satisfaction and distributed reasoning research. Distributed CSP frameworks formalize settings in which agents possess only partial views of the global constraint theory and must coordinate under incomplete observability [13]. More recent LLM-oriented work extends these ideas into distributed reasoning benchmarks. HiddenBench demonstrates that collective reasoning performance degrades

substantially under distributed information settings, where agents over-focus on shared evidence while neglecting private information unavailable to other agents[14, 15]. Similarly, PIANIST models partially observable multi-agent planning through explicit information partition functions and structured world models, emphasizing that naive LLM-as-planner approaches struggle under hidden-state interaction complexity [16]. These findings motivate evaluating orchestration quality under strict partition fidelity rather than unrestricted semantic cooperation between agents.

One possible way to expose orchestration weaknesses more clearly is through symbolic execution frameworks in which agent behavior is constrained by explicit local theories and deterministic reasoning procedures [17]. Unlike unconstrained LLM execution, symbolic execution agents operate over locally verifiable reasoning steps, making synchronization assumptions and orchestration inconsistencies easier to detect. Prior neuro-symbolic systems similarly use symbolic control and verification layers to constrain execution behavior and improve reasoning traceability [18, 19]. These systems suggest that symbolic-faithful execution can serve as a useful stress-test for evaluating whether orchestration remains robust under strict local reasoning constraints. Recent work also argues that raw task completion metrics are insufficient for evaluating coordination quality in LLM-MAS [20, 21, 22]. Procedure-Aware Evaluation (PAE) demonstrates that many apparent benchmark "successes" are achieved through integrity violations and policy-breaking behavior[23, 24]. Other work further shows that hidden coordination channels and unintended information propagation can undermine evaluation validity in multi-agent systems [25, 26]. Collectively, these findings motivate treating role fidelity and constraint adherence as first-class evaluation targets rather than relying exclusively on solve success metrics.

However, existing evaluations rarely jointly study strict horizontal information partitioning, centralized LLM orchestration, symbolic-faithful execution agents, and explicit role-compliance evaluation within a controlled coordination benchmark. Consequently, current LLM-MAS evaluations may overestimate orchestration capability when coordination success is not jointly evaluated with partition-faithful reasoning behavior. This work addresses this gap through a controlled benchmark that isolates coordination stagnation and role-compliance tradeoffs under horizontally partitioned reasoning.

3. Experimental Framework

This section presents the experimental framework used to evaluate coordination under strict horizontal reasoning partitioning. The framework is designed as a controlled coordination task in which multiple agents operate under restricted local theories while a centralized planner attempts to resolve coordination stagnation through iterative orchestration. Illustrated in Figure 1. Appendix A contains the full puzzle along with the solution.

3.1. Task Definition

We formulate the benchmark as a horizontally partitioned constraint satisfaction problem (CSP) inspired by zebra-style logical reasoning tasks. The environment consists of four houses and sixteen attribute variables distributed across multiple domains. The objective is to recover the unique globally consistent assignment through iterative coordination between specialized agents. Unlike standard centralized CSP solving, no execution agent has access to the complete constraint set. Instead, each agent operates only on a partitioned local theory corresponding to its assigned reasoning role. Consequently, progress depends on successful orchestration and information integration across partial reasoning views. The benchmark is intentionally constructed to induce coordination stagnation under incomplete local observability while remaining fully solvable under the complete global theory.

Formally, let:

$$H = \{h_1, h_2, h_3, h_4\}$$

be the ordered set of houses, where h_1 is the leftmost house and h_4 the rightmost. Each house is assigned exactly one value from each attribute domain:

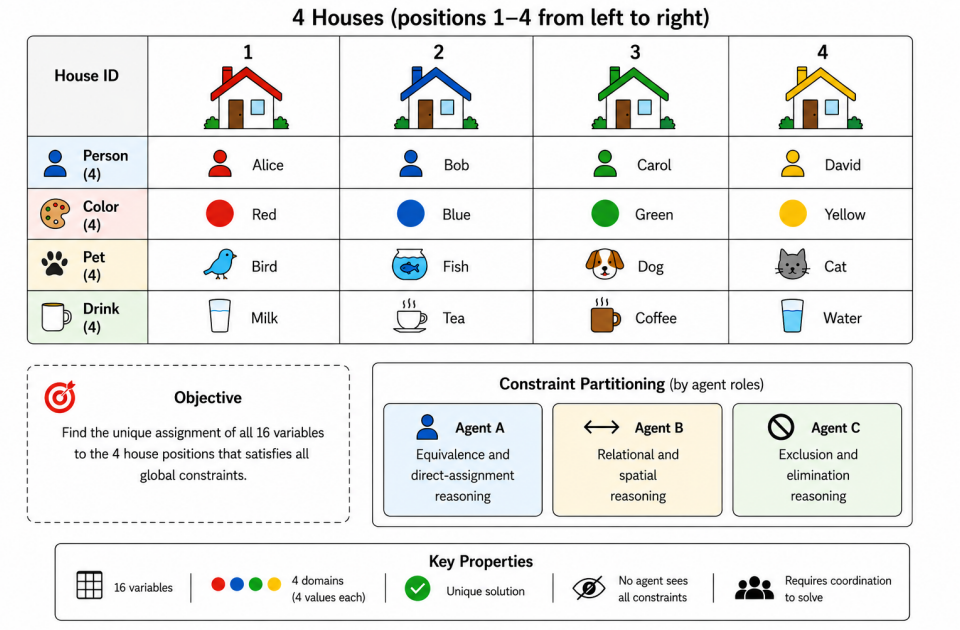

Figure 1: Puzzle Overview

$$P = \{\text{Alice}, \text{Bob}, \text{Carol}, \text{David}\}$$

$$C = \{\text{Red}, \text{Blue}, \text{Green}, \text{Yellow}\}$$

$$T = \{\text{Bird}, \text{Cat}, \text{Dog}, \text{Fish}\}$$

$$D = \{\text{Milk}, \text{Tea}, \text{Coffee}, \text{Water}\}$$

The benchmark uses the predicates:

$$\text{Lives}(p, h),\ \ \text{Color}(h, c),\ \ \text{Pet}(h, t),\ \ \text{Drink}(h, d)$$

together with relational predicates:

$$\text{RightOf}(h_i, h_j), \qquad \text{NextTo}(h_i, h_j)$$

where $\text{RightOf}(h_i, h_j)$ denotes immediate right adjacency and $\text{NextTo}(h_i, h_j)$ denotes neighboring houses. Global uniqueness constraints enforce bijective assignments between houses and attribute values:

$$\forall h \forall v_1 \forall v_2 \ : \ (A(h, v_1) \wedge A(h, v_2)) \rightarrow v_1 = v_2$$

$$\forall h_1 \forall h_2 \forall v \ : \ (A(h_1, v) \wedge A(h_2, v)) \rightarrow h_1 = h_2$$

for each attribute family $A \in \{\text{Color}, \text{Pet}, \text{Drink}\}$, with analogous constraints for Lives. The benchmark additionally includes shared ground facts visible to all agents:

$$\text{Pet}(h_1, \text{Bird})$$

$$\text{Lives}(\text{Bob}, h_2) \wedge \text{Color}(h_2, \text{Blue}) \wedge \text{Drink}(h_2, \text{Tea})$$

$$\text{Lives}(\text{Carol}, h_3) \wedge \text{Drink}(h_3, \text{Coffee})$$

The complete theory admits exactly one globally consistent solution.

3.2. Horizontal Constraint Partitioning

The global constraint theory is partitioned horizontally across three execution agents:

- **Agent A**: equivalence and direct-assignment reasoning,
- **Agent B**: relational and spatial reasoning,
- **Agent C**: exclusion and elimination reasoning.

Each agent receives only the subset of constraints assigned to its partition. Agents do not directly communicate with one another and cannot access hidden constraints outside their local theory. Formally, Agent A receives equivalence constraints of the form:

$$\forall h \ : \ \text{Lives}(\text{Alice}, h) \leftrightarrow \text{Color}(h, \text{Red})$$

$$\forall h \ : \ \text{Drink}(h, \text{Milk}) \leftrightarrow \text{Pet}(h, \text{Bird})$$

Agent B receives relational constraints:

$$(\text{Color}(h_\ell, \text{Blue}) \land \text{Color}(h_r, \text{Green})) \rightarrow \text{RightOf}(h_r, h_\ell)$$

$$(\text{Lives}(\text{Bob}, h_b) \land \text{Pet}(h_d, \text{Dog})) \rightarrow \text{NextTo}(h_b, h_d)$$

Agent C receives exclusion constraints:

$$\neg \, \text{Lives}(\text{David}, h_1)$$

$$\text{Pet}(h, \text{Fish}) \rightarrow \neg \, \text{Color}(h, \text{Yellow})$$

No individual agent theory is sufficient to uniquely solve the benchmark independently. The puzzle becomes solvable only after information derived from multiple partitions is integrated through the centralized orchestration process.

The shared state consists only of the global candidate memory maintained by the orchestration loop. This memory contains current candidate sets, resolved assignments, and previously accepted eliminations. The centralized planner observes the shared memory summary but does not receive the full hidden benchmark specification or the complete unconstrained clue set.

3.3. Orchestration Framework

The orchestration loop consists of a centralized planner and three execution agents operating over shared candidate memory. At each round, the planner selects the next execution agent, the target variable, and the coordination action to perform. The planner prompt includes the following.

1. the current round index,
2. the stagnation counter,
3. a textual summary of the shared candidate memory,
4. and the most recent planner decisions.

The planner does not receive direct access to hidden partitioned constraints beyond their indirect effect on the shared candidate state. Execution agents return structured outputs consisting of candidate eliminations, newly inferred singleton facts, optional reasoning traces. After each iteration, the shared memory is updated and evaluated for progress.

3.4. Experimental Conditions

We evaluate three primary execution conditions:

1. LLM backend with natural-language prompting,
2. LLM backend with Prolog-style prompting, and
3. Symbolic Prolog backend.

The natural-language configuration presents agent-local constraints as textual reasoning instructions. The Prolog-style prompting configuration represents the same local theory using structured symbolic-style constraints while preserving the same output interface. The symbolic backend executes deterministic symbolic reasoning over the same partitioned local theories. Given a fixed query, local domain state, and partition specification, the symbolic execution step is deterministic.

Additional mitigation-oriented variants, including plan-review and neuro-symbolic verification configurations, are not part of the primary experimental comparison and are discussed separately as exploratory extensions.

3.5. Progress and Coordination Stagnation

Let s_r denote the stagnation counter after round r, with $s_0 = 0$. After each iteration:

$$s_r = \begin{cases} 0 & \text{if memory progress occurs} \\ s_{r-1} + 1 & \text{otherwise} \end{cases}$$

Coordination stagnation occurs when:

$$s_r \geq K_{\text{stop}}$$

where K_{stop} denotes the configured stagnation threshold.

Memory progress is defined as either a change in the shared memory fingerprint, or a strict increase in the number of singleton assignments. The memory fingerprint is computed from a canonical JSON snapshot containing candidate sets and resolved assignments.

A separate lower threshold is used internally to trigger replanning behavior within the orchestration loop. Replanning therefore acts as a recovery mechanism, whereas coordination stagnation denotes sustained non-progress termination.

3.6. Role Compliance Evaluation

To evaluate partition fidelity, we introduce role compliance as a first-class evaluation metric.

Role compliance is computed through replay validation over the logged execution trace. Each structured output produced by Agents A, B, and C is validated against the corresponding partitioned local theory.

Compliance evaluation operates at the level of individual structured actions, including candidate eliminations and singleton fact assertions. An elimination is marked as a violation if the removed value is no longer present in the active candidate set, or the elimination is still feasible under the agent's local partitioned theory.

Similarly, singleton fact assertions are marked as violations if the asserted assignment is infeasible under the corresponding local theory.

The overall compliance rate is computed as:

$$\text{Compliance Rate} = \frac{\text{checks}_{\text{total}} - \text{violations}_{\text{total}}}{\text{checks}_{\text{total}}}$$

In addition to structural validation, we separately track orchestration leaks using rule-based detection over free-text reasoning traces. These leaks capture planner-like coordination behavior such as strategy recommendations, replanning suggestions, or meta-coordination language. Orchestration leaks are analyzed separately from the primary compliance metric.

3.7. Experimental Structure and Evaluation Metrics

The evaluation follows a mixed experimental structure consisting of representative single-run execution traces for mechanistic analysis and aggregate sweep-level statistics across experimental conditions. The primary analysis focuses on controlled execution traces to study planner behavior, coordination stagnation, and partition-boundary violations under strict horizontal reasoning partitioning. The experimental comparison reports the following metrics:

- **Termination Reason**: final execution outcome,
- **Progress Rounds**: number of rounds producing memory progress,
- **Role Compliance**: overall compliance rate,
- **Exact Match**: whether all slots match the gold solution.

Secondary diagnostic metrics, including orchestration leaks and planner trace statistics, are analyzed qualitatively in the failure analysis section.

4. Results

This section evaluates the behavior of LLM-based orchestration under strict horizontal reasoning partitioning using the ABC benchmark described in Section3. Across all experiments, the orchestration loop, planner configuration, shared memory structure, and agent partitioning remain fixed. The primary experimental variable is the execution backend used by the specialized agents.

We evaluate three execution settings: (1) LLM execution agents with natural-language prompting, (2) LLM execution agents with Prolog-style prompting, and (3) Symbolic Prolog execution agents. In all three settings, the centralized planner is implemented as an LLM-based orchestrator operating under the same coordination loop. Agents A, B, and C retain the same reasoning responsibilities and partial observability constraints across all conditions.

The analysis focuses on three questions:

1. *whether LLM planners can resolve coordination stagnation under strict horizontal partitioning*,
2. *why orchestration succeeds more reliably in LLM-only execution settings*, and
3. *whether successful coordination correlates with reduced role/persona compliance*.

4.1. Main Comparative Results

Table 1 summarizes the primary experimental outcomes across the evaluated execution settings.

Table 1
Main comparative results under horizontal reasoning partitioning.

Condition	Termination Reason	Rounds	Role Compliance	Exact Match
LLM + NL	PuzzleSolved	13	0.8889	Yes
LLM + Prolog	PuzzleSolved	14	0.8095	Yes
Symbolic Backend	Stagnation	5	1.0	No

The results show a clear behavioral difference between LLM-based execution agents and symbolic execution agents under identical orchestration conditions. Both LLM execution settings successfully solved the benchmark puzzle and achieved exact agreement with the gold CSP solution. In contrast, the symbolic execution backend consistently terminated through coordination stagnation despite maintaining perfect role compliance.

The symbolic backend achieved the highest compliance score (1.0) because all eliminations and candidate updates were constrained by deterministic local feasibility checks over the partitioned

theories. However, despite the correctness of local symbolic reasoning, the planner repeatedly failed to recover meaningful global progress once local deductions plateaued.

By contrast, the LLM execution conditions generated substantially more progress rounds and eventually completed the puzzle. The natural-language condition achieved the highest compliance among the successful configurations (0.8889), while the Prolog-style condition solved the puzzle with a lower compliance rate (0.8095). These results suggest that successful orchestration under horizontal partitioning does not necessarily imply strict partition-faithful reasoning.

More importantly, the results support the central hypothesis of this work: LLM planners coordinate LLM execution agents more successfully than symbolic execution agents under the same incomplete-observability setting. The symbolic backend exposes orchestration limitations that remain partially hidden in LLM-only execution settings.

4.2. Coordination Stagnation Analysis

Table 2 showcases the coordination stagnation behavior across execution conditions.

Table 2
Coordination stagnation characteristics across execution conditions.

Condition	Rounds Executed	Stagnation Triggered	Final Progress State
LLM + NL	48	No	Puzzle solved (13 progress rounds)
LLM + Prolog	40	No	Puzzle solved (14 progress rounds)
Symbolic Backend	31	Yes	Stagnated after 5 progress rounds

The symbolic execution condition terminated after 31 rounds with only 5 progress rounds before entering sustained non-progress execution. Planner traces showed repeated querying patterns over similar target variables and agent selections without generating sufficiently informative updates to produce additional singleton assignments or effective candidate eliminations.

Importantly, the symbolic agents themselves remained locally valid throughout execution. The observed failure therefore emerged primarily at the orchestration level rather than from incorrect symbolic deduction. The planner repeatedly cycled through locally valid but globally unproductive coordination actions, indicating ineffective replanning behavior under strict partition fidelity.

In contrast, both LLM execution conditions avoided terminal coordination stagnation and eventually solved the puzzle. The natural-language condition required more execution rounds and exhibited occasional stagnation-triggered replanning and hard pivots, suggesting less stable coordination dynamics. The Prolog-style condition solved the benchmark in fewer rounds while also achieving slightly more progress rounds, indicating that structured prompting improved orchestration efficiency under the same planner configuration.

These findings support the interpretation that coordination stagnation in the symbolic setting is not caused by insufficient local reasoning capability. Instead, the failure emerges from the planner's inability to strategically compose distributed partial inferences into sustained global progress under strict partition-restricted execution.

4.3. Role Compliance and Partition-Boundary Violations

To evaluate whether successful coordination preserves strict partition fidelity, we analyze role/persona compliance across execution conditions. Table 3 summarizes the compliance analysis.

The symbolic backend achieved perfect compliance with zero unsupported eliminations or invalid actions. This behavior is expected because all symbolic updates are generated through deterministic feasibility checks over explicitly partitioned local theories. However, despite perfect role fidelity, the symbolic execution setting failed to solve the puzzle. In contrast, both LLM execution conditions achieved exact puzzle resolution while simultaneously producing measurable partition-boundary violations.

Table 3
Role compliance and partition-boundary violation analysis.

Condition	Compliance Rate	Unsupported Eliminations	Invalid Assertions	Orchestration Leaks
LLM + NL	0.8889	4	—	0
LLM + Prolog	0.8095	12	—	0
Symbolic Backend	1.0	0	—	0

The natural-language condition produced four unsupported eliminations, while the Prolog-style condition produced twelve unsupported eliminations. Unsupported eliminations correspond to actions that cannot be justified solely from the agent's visible local theory during replay validation. These violations therefore indicate reasoning behavior that exceeds the explicitly assigned partition constraints. This result is central to the paper's second hypothesis. The experiments suggest that the stronger coordination performance observed in LLM-only execution settings may partially depend on role leakage and unjustified inference behavior rather than purely robust long-horizon orchestration.

Interestingly, the Prolog-style prompting condition exhibited lower compliance than the natural-language condition despite exposing more explicit structured constraints to the execution agents. This result suggests that structured symbolic-style prompting alone is insufficient to guarantee partition-faithful reasoning behavior.

The compliance analysis further showed that all violations in the natural-language configuration originated from Agent B, the relational and spatial reasoning agent. This observation is consistent with the broader difficulty of relational coordination under partial observability.

Notably, orchestration leak counts remained zero across all evaluated conditions. This indicates that the observed violations primarily emerged through unsupported eliminations rather than explicit planner-like coordination language or overt role drift in the generated reasoning traces.

4.4. Prompting Style Comparison

Table 4 compares natural-language and Prolog-style prompting for the LLM execution conditions.

Table 4
Effect of prompting style on coordination and compliance.

Prompt Style	Exact Match	Role Compliance	Termination Reason
Natural Language	Yes	0.8889	PuzzleSolved
Prolog-Style	Yes	0.8095	PuzzleSolved

Both prompting conditions achieved exact puzzle resolution, demonstrating that structured symbolic-style prompting is not strictly necessary for successful coordination in this benchmark. However, the two conditions exhibited different tradeoffs between coordination efficiency and partition fidelity. The Prolog-style condition completed the benchmark in fewer rounds and achieved slightly more progress rounds, but also produced substantially more unsupported eliminations and lower overall compliance.

These findings suggest that exposing structured symbolic-style constraints to LLM execution agents does not eliminate unjustified inference behavior. Instead, prompt structure appears to influence how aggressively execution agents generate eliminations and candidate reductions during coordination.

4.5. C0/C1/C2 Symbolic Ablations

To further analyze the symbolic execution setting, we evaluated three symbolic ablation configurations. Table 5 summarizes the findings of the abltaion study.

All three ablation configurations terminated through coordination stagnation while maintaining perfect role compliance. Neither singleton propagation parity (C1) nor widened multi-slot querying (C2) enabled successful puzzle completion.

Table 5
C0/C1/C2 ablation analysis.

Configuration	Termination Reason	Progress Rounds	Role Compliance	Exact Match
C0	CoordinationStagnation	5	1.0	No
C1	CoordinationStagnation	5	1.0	No
C2	CoordinationStagnation	3	1.0	No

These results suggest that the primary limitation lies in orchestration-level coordination rather than local symbolic deduction granularity. Even under deterministic, fully role-faithful execution, the planner remained unable to reliably recover progress once distributed reasoning dependencies became sufficiently constrained.

5. Conclusion

This paper investigated the reliability of LLM-based orchestration under strict horizontal knowledge partitioning in heterogeneous multi-agent systems. Using a distributed zebra-style constraint satisfaction benchmark, we evaluated whether an LLM planner could coordinate execution agents operating under incomplete local observability and partition-restricted reasoning roles.

Across all experiments, the orchestration framework, planner configuration, and agent partitioning remained fixed, while the execution backend varied between LLM-based and symbolic Prolog-style agents. The results revealed a consistent behavioral difference between these settings. LLM execution agents successfully solved the benchmark task under both natural-language and Prolog-style prompting, whereas the symbolic execution backend repeatedly entered coordination stagnation despite maintaining perfect role compliance and locally valid reasoning. These findings suggest that current LLM planners struggle to reliably coordinate symbolic or strictly role-faithful agents under horizontally partitioned reasoning constraints.

The compliance analysis further showed that successful LLM-only coordination frequently coincided with unsupported eliminations and partition-boundary violations. In contrast, the symbolic backend remained fully compliant but failed to sustain meaningful global progress. This tradeoff suggests that apparent coordination success in LLM-only MAS may partially depend on unjustified inference behavior and implicit cross-partition reasoning rather than robust long-horizon orchestration alone. Consequently, solve success by itself may be insufficient for evaluating coordination quality in LLM-based multi-agent systems.

More broadly, this work highlights coordination stagnation as a failure mode of LLM orchestration under strict horizontal partitioning. The experiments demonstrate that orchestration quality should be evaluated jointly with partition fidelity and role compliance rather than relying exclusively on task completion metrics. Future work should investigate larger distributed reasoning benchmarks, stronger symbolic-faithful execution settings, and verification-aware orchestration mechanisms capable of preserving both coordination effectiveness and strict role-local reasoning validity.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT for sentence polishing and rephrasing. After using it, the authors reviewed and edited the content and take full responsibility for the content of the publication.

References

[1] X. Li, S. Wang, S. Zeng, Y. Wu, Y. Yang, A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges, Vicinagearth 1 (2024). doi:10.1007/s44336-024-00009-2.

[2] R. Aratchige, D. W. Ilmini, Llms working in harmony: A survey on the technological aspects of building effective llm-based multi agent systems, ArXiv abs/2504.01963 (2025). doi:10.48550/arxiv.2504.01963.

[3] S. Han, Q. Zhang, Y. Yao, W. Jin, Z. Xu, C. He, Llm multi-agent systems: Challenges and open problems, ArXiv abs/2402.03578 (2024). doi:10.48550/arxiv.2402.03578.

[4] M. Cemri, M. Z. Pan, S. Yang, L. A. Agrawal, B. Chopra, R. Tiwari, K. Keutzer, A. G. Parameswaran, D. Klein, K. Ramchandran, M. Zaharia, J. E. Gonzalez, I. Stoica, Why do multi-agent llm systems fail?, ArXiv abs/2503.13657 (2025). doi:10.48550/arxiv.2503.13657.

[5] E. Y. Chang, L. Geng, Alas: A stateful multi-llm agent framework for disruption-aware planning, Multi-LLM Agent Collaborative Intelligence (2025). doi:10.1145/3749421.3749436.

[6] S. Raza, R. Sapkota, M. Karkee, C. Emmanouilidis, Trism for agentic ai: A review of trust, risk, and security management in llm-based agentic multi-agent systems, ArXiv abs/2506.04133 (2025). doi:10.48550/arxiv.2506.04133.

[7] V. Vinay, Failure modes in llm systems: A system-level taxonomy for reliable ai applications, ArXiv abs/2511.19933 (2025). doi:10.48550/arxiv.2511.19933.

[8] W. Zhang, C. Cui, Y. Zhao, R. Hu, Y. Liu, Y. Zhou, B. An, Agentorchestra: A hierarchical multi-agent framework for general-purpose task solving, ArXiv abs/2506.12508 (2025). doi:10.48550/arxiv.2506.12508.

[9] Z. Wang, F. Wu, H. Wang, X. Tang, B. Li, Z. Yin, Y. Ma, Y. Li, W. Sun, X. Chen, Y. Ye, Why reasoning fails to plan: A planning-centric analysis of long-horizon decision making in llm agents, ArXiv abs/2601.22311 (2026). doi:10.48550/arxiv.2601.22311.

[10] M. Samiei, M. Mansouri, M. Baghshah, The illusion of procedural reasoning: Measuring long-horizon fsm execution in llms, ArXiv abs/2511.14777 (2025). doi:10.48550/arxiv.2511.14777.

[11] H. Li, Y. Q. Chong, S. Stepputtis, J. Campbell, D. Hughes, M. Lewis, K. P. Sycara, Theory of mind for multi-agent collaboration via large language models, 2023, pp. 180–192. doi:10.18653/v1/2023.emnlp-main.13.

[12] W. Chen, S. Koenig, B. Dilkina, Why solving multi-agent path finding with large language model has not succeeded yet, ArXiv abs/2401.03630 (2024). doi:10.48550/arxiv.2401.03630.

[13] F. Fioretto, E. Pontelli, W. Yeoh, Distributed constraint optimization problems and applications: A survey, ArXiv abs/1602.06347 (2016). doi:10.1613/jair.5565.

[14] Y. Li, A. Naito, H. Shirado, Systematic failures in collective reasoning under distributed information in multi-agent llms (2025).

[15] W. Liu, C. Wang, Y. Wang, Z. Xie, R. Qiu, Y. Dang, Z. Du, W. Chen, C. Yang, C. Qian, Autonomous agents for collaborative task under information asymmetry, ArXiv abs/2406.14928 (2024). doi:10.48550/arxiv.2406.14928.

[16] J. Light, S. Xing, Y. Liu, W. Chen, M. Cai, X. Chen, G. Wang, W. Cheng, Y. Yue, Z. Hu, Pianist: Learning partially observable world models with llms for multi-agent decision making, arXiv preprint arXiv:2411.15998 (2024).

[17] V. Belle, M. Fisher, A. Russo, E. Komendantskaya, A. Nottle, Neuro-symbolic ai + agent systems: A first reflection on trends, opportunities and challenges (2024) 180–200. doi:10.1007/978-3-031-56255-6_10.

[18] B. Callewaert, S. Vandevelde, J. Vennekens, Verus-lm: a versatile framework for combining llms with symbolic reasoning, 2025, pp. 47–62. doi:10.4204/eptcs.439.5.

[19] M. Kim, Bridging symbolic control and neural reasoning in llm agents: The structured cognitive loop, ArXiv abs/2511.17673 (2025). doi:10.48550/arxiv.2511.17673.

[20] K. Zhu, H. Du, Z. Hong, X. Yang, S. Guo, Z. Wang, Z. Wang, C. Qian, X. Tang, H. Ji, J. You, Multiagentbench: Evaluating the collaboration and competition of llm agents, ArXiv abs/2503.01935 (2025). doi:10.48550/arxiv.2503.01935.

[21] J. Lee, R. Chang, D. Kwon, H. Singh, N. Verma, Gemmas: Graph-based evaluation metrics for multi agent systems, ArXiv abs/2507.13190 (2025). doi:10.48550/arxiv.2507.13190.

[22] S. Akshathala, B. Adnan, M. Ramesh, K. Vaidhyanathan, B. Muhammed, K. Parthasarathy, Beyond task completion: An assessment framework for evaluating agentic ai systems, ArXiv abs/2512.12791 (2025). doi:10.48550/arxiv.2512.12791.

[23] H. Cao, I. Driouich, E. Thomas, Beyond task completion: Revealing corrupt success in llm agents through procedure-aware evaluation (2026).

[24] C. Zheng, Y. Cao, X. Dong, T. He, Demonstrations of integrity attacks in multi-agent systems, ArXiv abs/2506.04572 (2025). doi:10.48550/arxiv.2506.04572.

[25] S. Motwani, M. Baranchuk, M. Strohmeier, V. Bolina, P. Torr, L. Hammond, C. S. de Witt, Secret collusion among ai agents: Multi-agent deception via steganography, Advances in Neural Information Processing Systems 37 (2024). doi:10.52202/079017-2336.

[26] Y. Mathew, O. Matthews, R. McCarthy, J. Velja, C. S. de Witt, D. Cope, N. Schoots, Hidden in plain text: Emergence mitigation of steganographic collusion in llms (2024) 585–624. doi:10.48550/arxiv.2410.03768.

Appendix

A. Formal Puzzle Solution Walkthrough

This appendix presents the formal step-by-step derivation for the easy benchmark instance. The puzzle definition, predicates, global axioms, and horizontally partitioned constraints are given in the main text.

A.1. Benchmark Pins

The following ground literals are shared by all agents:

$$\text{Pet}(h_1, \text{Bird})$$

$$\text{Lives}(\text{Bob}, h_2) \wedge \text{Color}(h_2, \text{Blue}) \wedge \text{Drink}(h_2, \text{Tea})$$

$$\text{Lives}(\text{Carol}, h_3) \wedge \text{Drink}(h_3, \text{Coffee})$$

A.2. Agent Constraints

Agent A Same-house equivalences

$$\forall h.\ \big(\text{Lives}(\text{Alice}, h) \leftrightarrow \text{Color}(h, \text{Red})\big)$$

$$\forall h.\ \big(\text{Drink}(h, \text{Milk}) \leftrightarrow \text{Pet}(h, \text{Bird})\big)$$

Agent B Relational constraints

$$\forall h_\ell \forall h_r.\ \big(\text{Color}(h_\ell, \text{Blue}) \wedge \text{Color}(h_r, \text{Green})\big) \rightarrow \text{RightOf}(h_r, h_\ell)$$

$$\forall h_b \forall h_d.\ \Big(\text{Lives}(\text{Bob}, h_b) \wedge \text{Pet}(h_d, \text{Dog}) \wedge h_b \neq h_d\Big) \rightarrow \text{NextTo}(h_b, h_d)$$

Agent C Exclusion constraints

$$\neg\, \text{Lives}(\text{David}, h_1)$$

$$\forall h.\ \big(\text{Pet}(h, \text{Fish}) \rightarrow \neg\, \text{Color}(h, \text{Yellow})\big)$$

A.3. Cooperative Derivation

All agents pool their knowledge together with the shared benchmark pins and global uniqueness axioms.

1. From the benchmark pins:

$$\text{Pet}(h_1, \text{Bird})$$

$$\text{Lives}(\text{Bob}, h_2), \quad \text{Color}(h_2, \text{Blue}), \quad \text{Drink}(h_2, \text{Tea})$$

$$\text{Lives}(\text{Carol}, h_3), \quad \text{Drink}(h_3, \text{Coffee})$$

2. Using Agent A's equivalence:

$$\text{Drink}(h, \text{Milk}) \leftrightarrow \text{Pet}(h, \text{Bird})$$

and

$$\text{Pet}(h_1, \text{Bird})$$

infer:

$$\text{Drink}(h_1, \text{Milk})$$

3. From:

$$\text{Color}(h_2, \text{Blue})$$

together with Agent B's immediate-right constraint:

$$\text{Blue} \rightarrow \text{Green immediately right}$$

infer:

$$\text{Color}(h_3, \text{Green})$$

since h_3 is immediately to the right of h_2.

4. Using Agent A's equivalence:

$$\text{Lives}(\text{Alice}, h) \leftrightarrow \text{Color}(h, \text{Red})$$

observe that:

$$h_2 \neq \text{Red}$$

because h_2 is Blue, and

$$h_3 \neq \text{Red}$$

because h_3 is Green.
Therefore Red must be either h_1 or h_4.
Since h_3 is occupied by Carol and h_2 by Bob, Alice must occupy either h_1 or h_4. By uniqueness of persons and colors:

$$\text{Lives}(\text{Alice}, h_1)$$

and

$$\text{Color}(h_1, \text{Red})$$

5. By uniqueness of persons, the remaining unassigned person for h_4 is David:

$$\text{Lives}(\text{David}, h_4)$$

which is also consistent with:

$$\neg\, \text{Lives}(\text{David}, h_1)$$

6. By uniqueness of colors, the remaining color for h_4 is Yellow:

$$\text{Color}(h_4, \text{Yellow})$$

7. From:

$$\text{Lives}(\text{Bob}, h_2)$$

and Agent B's adjacency constraint:

$$\text{Bob lives next to the Dog owner}$$

the Dog must be located at either h_1 or h_3.
Since:

$$\text{Pet}(h_1, \text{Bird})$$

the Dog cannot be at h_1. Therefore:

$$\text{Pet}(h_3, \text{Dog})$$

8. By uniqueness of pets, the remaining pets for h_2 and h_4 are Fish and Cat.
Agent C states:

$$\text{Fish} \nleftrightarrow \text{Yellow}$$

and since:

$$\text{Color}(h_4, \text{Yellow})$$

infer:

$$\text{Pet}(h_4, \text{Cat})$$

and

$$\text{Pet}(h_2, \text{Fish})$$

9. By uniqueness of drinks:

$$\text{Drink}(h_1, \text{Milk}), \quad \text{Drink}(h_2, \text{Tea}), \quad \text{Drink}(h_3, \text{Coffee})$$

therefore the remaining drink for h_4 is:

$$\text{Drink}(h_4, \text{Water})$$

A.4. Final Solution

Table 6
Unique Solution for the Easy Benchmark

House	Person	Color	Pet	Drink
h_1	Alice	Red	Bird	Milk
h_2	Bob	Blue	Fish	Tea
h_3	Carol	Green	Dog	Coffee
h_4	David	Yellow	Cat	Water

The resulting assignment satisfies all benchmark pins, uniqueness axioms, topology constraints, and all agent-local theories simultaneously. By exhaustion of all remaining domains under the global constraints, the solution is unique.