Multi-Agent Cooperative Planning and Execution Framework for Distributed LLM Reasoning Systems

Bastian J. Page
Department of Computer Science, University of North Texas, Denton, TX, USA.
contactbastian@unt.edu

Harish Mahajan
School of Information Technology, University of Cincinnati, Cincinnati, OH, USA.
harishmail@uc.edu

Xuan Cai
Department of Computer Science, University of Alabama at Birmingham, Birmingham, AL, USA.
cai1970@uab.edu

Hugo Sanders
Department of Computer Science, University of Houston, Houston, TX, USA.
hugo643@uh.edu

Abstract

The rapid scaling of large language models has introduced significant challenges in reasoning coherence, computational efficiency, and operational robustness when deployed in real-world, distributed environments. This paper proposes a comprehensive multi-agent cooperative planning and execution framework designed to address these challenges by decomposing complex reasoning tasks into subtasks that are allocated across a network of specialized LLM agents. The framework integrates hierarchical planning with decentralized execution, enabling agents to dynamically coordinate through structured communication protocols and shared memory architectures. Emphasis is placed on structural trade-offs between centralized orchestration and autonomous agent decision-making, including considerations of latency, fault tolerance, and alignment. The paper further examines governance mechanisms for ensuring fairness and accountability in multi-agent systems, as well as infrastructure requirements for sustainable deployment at scale. A case illustration involving a distributed medical diagnosis system demonstrates the practical applicability of the proposed architecture. The discussion extends to policy implications, including regulatory frameworks for agent oversight and data sovereignty. By synthesizing insights from distributed systems, artificial intelligence, and socio-technical infrastructure research, this work contributes a systems-level perspective on the design of cooperative LLM reasoning platforms that are both performant and ethically grounded.

Keywords

multi-agent systems, large language models, distributed reasoning, cooperative planning, execution framework, system governance, socio-technical infrastructure.

1. Introduction

The emergence of large language models has transformed the landscape of natural language understanding and generation, enabling unprecedented capabilities in tasks ranging from code synthesis to complex reasoning. However, as these models are deployed in distributed, multi-user environments, their inherent limitations in handling long-range dependencies, maintaining context across sessions, and ensuring factual consistency become pronounced. A single monolithic LLM, even with massive parameter counts, struggles to efficiently manage the combinatorial complexity of real-world workflows that require sequential reasoning, multi-step planning, and collaboration across knowledge domains. This has motivated the development of multi-agent architectures in which individual LLM instances, each with specialized roles or local knowledge bases, cooperate to solve problems that exceed the capacity of any single model [1], [2]. Such distributed reasoning systems offer potential improvements in scalability, robustness, and interpretability, but they also introduce new challenges in coordination, communication overhead, and alignment of agent objectives.

Contemporary research has explored the use of modular reasoning pipelines, where a central planner decomposes a high-level goal into subgoals and dispatches them to expert agents [3], [4]. A notable contribution in this direction is the Plan Then Action framework, which provides high-level planning guidance to reinforcement learning agents for improved reasoning [5]. This approach exemplifies a broader trend toward separating planning from execution, allowing agents to focus on targeted subproblems while a higher-level controller ensures global coherence. However, most existing works treat planning as a static, top-down process, whereas dynamic environments and task interdependencies demand adaptive, decentralized coordination. The framework proposed in this paper extends these ideas by integrating cooperative planning with execution in a distributed setting, where agents maintain local autonomy while participating in a shared reasoning process mediated by structured communication and conflict resolution mechanisms.

The motivation for a multi-agent cooperative framework stems from both practical and theoretical considerations. In practical deployments, such as automated customer support or collaborative scientific discovery, the reasoning tasks are inherently distributed across time, space, and expertise. A single model cannot realistically possess all domain knowledge or maintain the required bandwidth. Theoretically, multi-agent systems offer a natural lens through which to study emergent reasoning behaviors, resource allocation, and the trade-offs between centralized control and distributed autonomy. This paper adopts a systems engineering perspective, focusing on the architectural decisions that govern the performance, fairness, and sustainability of distributed LLM reasoning platforms.

2. Background and Related Work

The field of multi-agent systems has a rich history in artificial intelligence, with foundational work on distributed problem solving, negotiation protocols, and organizational structures [6]. With the advent of LLMs, these classical models have been revisited and adapted to leverage the generative and semantic capabilities of transformer-based architectures. Early attempts at multi-agent LLM systems employed simple voting or debating mechanisms to aggregate outputs from multiple models, improving factual accuracy and reducing hallucination [7]. More sophisticated approaches introduced role-based specialization, where agents receive different prompts or are fine-tuned for distinct functions such as retrieval, summarization, or verification [8]. These designs, while effective, often lack a coherent planning layer that can dynamically allocate tasks based on the evolving state of the reasoning process.

Research on planning for LLM reasoning has predominantly focused on single-agent strategies, including chain-of-thought prompting, tree-of-thought search, and self-consistency methods [9], [10]. These techniques operate within the representation space of a single model, limiting their ability to incorporate external knowledge or parallel computation. In contrast, distributed planning frameworks envision a separation between a planning agent and multiple execution agents, where the planner generates a high-level schedule and the executors carry out the steps. This line of work has been advanced by studies on hierarchical reinforcement learning and recursive reasoning [11], [12]. The required reference introduces a method that uses high-level planning guidance to improve the reasoning of a reinforcement learning agent [5], demonstrating that such hierarchical decomposition can significantly enhance performance on long-horizon tasks.

Another important stream of research concerns communication protocols in multi-agent LLM systems. Agents must exchange information efficiently without overwhelming the network or causing semantic drift. Some frameworks adopt a shared blackboard or common workspace where agents post intermediate results, while others rely on direct messaging with specified formats [13], [14]. The choice of communication modality influences both the speed and the reliability of the collective reasoning process. Additionally, the issue of agent alignment—ensuring that each agent's actions are consistent with the overall goal and with ethical constraints—has been addressed through value-guided design and reward shaping [15]. These considerations are particularly critical in domains such as healthcare, finance, and legal reasoning, where errors or biased outputs can have serious consequences.

3. Architectural Framework for Multi-Agent Cooperative Planning

The proposed framework comprises three primary layers: a global planning layer, a distributed execution layer, and a coordination infrastructure. The global planning layer consists of a set of planner agents, which may be LLM-based or rule-based, that decompose incoming queries into a directed acyclic graph of subtasks. Each subtask is assigned a priority level, a set of required capabilities, and expected input-output specifications. The planning process is itself iterative, allowing for replanning when an execution agent reports difficulty or when new information emerges during reasoning.

In the execution layer, multiple dedicated LLM agents operate in parallel, each equipped with a specialized prompt template, fine-tuned model, or external tool interface. These agents are not statically bound to roles; instead, they are dynamically selected and grouped based on the subtask requirements and current system load. This dynamic compositionality introduces flexibility but also necessitates a robust agent registry and capability description mechanism. The framework employs a service-oriented architecture where each agent advertises its skills, latency guarantees, and confidence metrics. The planner agents use this metadata to formulate an initial task allocation, which is then refined through a cooperative negotiation process. Negotiation involves agents bidding on subtasks, proposing modifications to the plan, and resolving resource conflicts via a priority-based arbitration protocol.

The coordination infrastructure includes a shared memory store that maintains the global reasoning state, including partial results, dependency graphs, and provenance records. This memory is accessible to all agents but with write permissions governed by a commit protocol to prevent inconsistent updates. The use of a distributed ledger or conflict-free replicated data type has been explored to ensure eventual consistency [16]. The coordination layer also monitors agent performance, detecting anomalies such as non-terminating loops, excessive token consumption, or divergence from the intended reasoning path. When an anomaly is

detected, the system can trigger a fallback mechanism, such as re-allocating the subtask to a different agent or escalating to a human supervisor.

One key architectural trade-off is between the granularity of subtask decomposition and the communication overhead. Finer-grained decomposition allows for more parallelism and specialization but increases the cost of coordination and potential for deadlock. Coarser-grained decomposition reduces overhead but may lead to underutilization of agent capabilities and longer sequential dependencies. The framework addresses this by introducing an adaptive granularity parameter that can be adjusted based on historical performance data and real-time metrics of network latency and agent availability.

4. Execution Dynamics and Coordination Mechanisms

Execution within the proposed framework proceeds through a series of rounds, each corresponding to the completion of a batch of subtasks. After the initial plan is generated, the planner emits a set of executable units, each assigned to one or more execution agents. These agents process the subtasks concurrently, periodically writing intermediate results to the shared memory. The coordination infrastructure maintains a dependency table that tracks which outputs are prerequisites for which pending subtasks. When a subtask completes, its results trigger the activation of dependent subtasks, propagating the reasoning forward.

To handle cases where multiple agents produce conflicting outputs for the same subtask—for instance, when two retrieval agents return different relevant documents—the framework incorporates a consensus mechanism. Agents may vote on the correctness or relevance of competing results, with votes weighted by each agent's historical reliability and confidence scores. Alternatively, a designated resolution agent uses a meta-reasoning step to synthesize a unified output, as demonstrated in recent ensemble methods [17]. This consensus step introduces latency but is essential for maintaining coherence and avoiding error propagation.

Another critical coordination mechanism is dynamic replanning. During execution, an agent may encounter a subtask that requires deeper hierarchical decomposition or that proves unsolvable given the agent's capabilities. The agent sends a feedback signal to the planner, which then revises the remaining portion of the plan. This closed-loop feedback aligns with the Plan Then Action paradigm, where planning is not a one-time event but an ongoing process guided by execution feedback [5]. The framework also supports opportunistic borrowing, where an agent that finishes its assigned work early can volunteer to help with pending subtasks from other agents, provided it has the necessary capabilities and that the reassignment does not violate dependency constraints.

The execution dynamics are further shaped by resource management policies. Each agent operates within a computational budget, measured in terms of API calls, inference time, or token limits. The framework employs a token banking system that allocates a fixed number of tokens per agent per reasoning cycle, with the ability to borrow from a shared reserve. This prevents any single agent from monopolizing resources and ensures fair access across agents. Monitoring tools track token consumption and flag potential exhaustion or starvation, triggering load balancing actions such as redistributing subtasks to underutilized agents.

5. Governance, Robustness, and Fairness Considerations

The deployment of multi-agent LLM reasoning systems in high-stakes domains necessitates robust governance frameworks to ensure that the collective decision-making process is transparent, accountable, and aligned with human values. Governance in this context refers to

the set of rules, oversight mechanisms, and audit trails that regulate agent behavior and system outcomes. A central challenge is the tension between agent autonomy and centralized control. Granting agents too much autonomy can lead to emergent behaviors that deviate from intended goals, while excessive control stifles the flexibility that makes multi-agent systems valuable.

One approach to governance is the implementation of a constitution-based system, where all agents are bound by a set of high-level principles that constrain their actions [18]. These principles may include directives to avoid harmful outputs, respect user privacy, and ensure factual accuracy. Agents are encouraged to self-enforce these rules, but external monitors periodically sample agent outputs and apply corrective penalties if violations are found. The monitors themselves are LLM-based but with conservative settings and a separate oversight chain to prevent collusion.

Robustness is another critical concern, particularly in the face of adversarial inputs, system failures, or distribution shifts. The framework employs redundancy and diversity in agent design to mitigate single points of failure. For example, multiple agents may be assigned to the same subtask using different prompts or different underlying model versions, and the final output is selected through a majority vote or a confidence-weighted aggregation. Additionally, the shared memory includes versioning and rollback capabilities, allowing the system to revert to a previous state if a reasoning path is found to be erroneous. The coordination infrastructure is designed with a circuit breaker pattern, where if the error rate exceeds a threshold, the system enters a safe mode that restricts agent autonomy and escalates all decisions to a human operator.

Fairness in multi-agent reasoning systems involves ensuring that the reasoning process does not systematically disadvantage certain user groups or propagate biases present in the training data. This requires careful attention to the composition of the agent pool. If agents are fine-tuned on skewed datasets, their outputs may reflect those biases. The framework incorporates bias auditing tools that analyze the distribution of agent outputs across demographic dimensions and trigger re-balancing interventions when disparities are detected. Furthermore, the planning layer can be designed to prioritize fairness objectives, for instance by requiring that certain subtasks be assigned to agents that have demonstrated equitable performance across different inputs [19]. Ethical considerations also extend to resource allocation; agents representing marginalized viewpoints must not be starved of computational budget simply because their reasoning paths are less common.

6. Infrastructure, Sustainability, and Deployment Challenges

Practical deployment of a multi-agent cooperative planning and execution framework requires a robust infrastructure that can support high-throughput, low-latency communication among distributed agents. Many real-world systems rely on cloud-based microservice architectures, with each agent deployed as a separate container or serverless function [20]. This design offers elasticity, allowing the system to scale the number of agents up or down based on demand. However, it also introduces challenges in network latency, especially when agents are geographically dispersed. The framework addresses this by placing agents close to the data sources they access and by using edge computing nodes for latency-sensitive subtasks.

Sustainability is a growing concern for large-scale AI systems. Running multiple LLM agents concurrently can consume enormous amounts of energy, contributing to carbon emissions. The framework incorporates energy-aware scheduling, where the planner considers the

energy efficiency of different models and hardware configurations when allocating subtasks. For example, a smaller distilled model might be preferred for simple retrievals, while a larger state-of-the-art model is reserved for complex reasoning steps. Dynamic voltage and frequency scaling techniques, as well as the use of specialized accelerators, can further reduce energy consumption [21]. The shared token banking system also indirectly promotes sustainability by discouraging wasteful token usage.

Deployment challenges include version management, model drift, and interoperability between different LLM providers. As models are updated, the behavior of individual agents may change, potentially disrupting the coordination protocols. The framework defines clear versioning conventions and compatibility tests that must be passed before a new agent version is allowed into the pool. Additionally, the system must handle cases where some agents are temporarily unavailable due to maintenance or network partitions. Graceful degradation strategies, such as fallback to a simpler reasoning mode or caching of previous results, ensure that the system continues to function, albeit with reduced performance.

Another dimension of deployment is the integration with existing enterprise systems. The multi-agent framework exposes standardized APIs that can be called by external applications, with authentication and rate limiting to prevent abuse. Data sovereignty and privacy regulations, such as the General Data Protection Regulation in the European Union, impose constraints on where and how data processed by agents can be stored and transferred. The framework supports data residency by allowing agents to be deployed in specific geographic regions and by encrypting all inter-agent communications.

7. Future Directions and Policy Implications

The development of multi-agent cooperative planning frameworks for distributed LLM reasoning is still in its early stages, and numerous avenues for future research remain open. One promising direction is the integration of reinforcement learning at the planning level, where the planner agent learns to optimize the decomposition and allocation strategies over time based on system performance feedback. The required work on high-level planning guidance for reinforcement learning offers a foundational method that could be extended to multi-agent settings [5]. Another area is the study of emergent cooperation and specialization among agents without explicit central coordination, drawing on insights from collective intelligence and evolutionary computation [22].

From a policy perspective, the governance of multi-agent AI systems raises important questions about accountability. When a reasoning system produces a harmful output, who is responsible—the planner, the execution agents, or the system operator? Current legal frameworks are ill-equipped to address the distributed nature of such systems. It is likely that future regulations will require documented audit trails, explainability mechanisms, and human-in-the-loop oversight for high-impact applications. The proposed framework's provenance logging and consensus mechanisms provide a foundation for compliance, but further work is needed to standardize these practices across industries.

Fairness and bias mitigation also have policy dimensions. Regulatory bodies may mandate that multi-agent reasoning systems be tested for disparate impact before deployment, similar to the way algorithms are evaluated in credit scoring and hiring. The framework's bias auditing and re-balancing capabilities can serve as a template for such evaluations. Additionally, there is a need for international standards on inter-agent communication

protocols and safety benchmarks, to ensure interoperability and trustworthiness across jurisdictional boundaries.

Finally, the sustainability of large-scale distributed LLM systems will increasingly come under scrutiny, both from an environmental standpoint and from the perspective of equitable access to computational resources. Policies that incentivize energy-efficient computing and provide funding for green AI research could accelerate the adoption of techniques like model distillation and carbon-aware scheduling. The framework's energy-aware scheduling module is a step in this direction, but broader systemic changes in hardware design and cloud infrastructure will be necessary.

8. Conclusion

This paper has presented a multi-agent cooperative planning and execution framework for distributed LLM reasoning systems, addressing the critical challenges of coordination, governance, robustness, and sustainability. By decomposing complex reasoning tasks into manageable subtasks and allocating them across a dynamic pool of specialized agents, the framework achieves greater scalability and flexibility than monolithic models. The integration of a hierarchical planning layer with decentralized execution, supported by shared memory and consensus mechanisms, enables adaptive and coherent reasoning in the presence of uncertainty and resource constraints. The discussion of governance mechanisms, including constitution-based enforcement and fairness auditing, highlights the importance of embedding ethical considerations into the architecture from the outset. Infrastructure and deployment challenges, particularly regarding latency, energy consumption, and regulatory compliance, were examined, with proposed solutions that balance performance with responsibility. As AI systems become more integral to decision-making in critical domains, the need for robust, transparent, and equitable multi-agent reasoning platforms will only grow. The framework outlined here provides a foundation for continued research and practical implementation, inviting collaboration across disciplines to refine and realize the vision of truly cooperative artificial intelligence.

References

1. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877-1901.

2. Wang, S., Li, K., & Yu, K. (2023). Multi-agent reinforcement learning for cooperative autonomous driving. IEEE Transactions on Intelligent Transportation Systems, 24(5), 5612-5624.

3. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35, 24824-24837.

4. Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., ... & Tsvetkov, Y. (2022). Least-to-most prompting enables complex reasoning in large language models. arXiv preprint arXiv:2205.10625.

5. Dou, Z., Zhao, Q., Wan, Z., Zhang, D., Wang, W., Raiyan, T., ... & Biswas, S. (2025). Plan Then Action: High-Level Planning Guidance Reinforcement Learning for LLM Reasoning. arXiv preprint arXiv:2510.01833.

6. Shoham, Y., & Leyton-Brown, K. (2009). Multiagent systems: Algorithmic, game-theoretic, and logical foundations. Cambridge University Press.

7. Du, Y., Li, S., Torralba, A., Tenenbaum, J., & Mordatch, I. (2023). Improving language models by retrieving from trillions of tokens. Proceedings of the 40th International Conference on Machine Learning, 8606-8621.

8. Park, J. S., O'Brien, J., Pope, R., & Anderson, M. (2023). Generative agents: Interactive simulacra of human behavior. Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, 1-15.

9. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., & Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. Advances in Neural Information Processing Systems, 36.

10. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., ... & Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171.

11. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

12. Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. Journal of Artificial Intelligence Research, 4, 237-285.

13. Li, G., Hammoud, H., Itani, H., Khizbullin, D., & Ghanem, B. (2023). Camel: Communicative agents for "mind" exploration of large language model society. Advances in Neural Information Processing Systems, 36.

14. Qian, C., Cong, X., Yang, C., Chen, W., Su, Y., Xu, J., ... & Chua, T. S. (2023). Communicative agents for software development. arXiv preprint arXiv:2307.07924.

15. Russell, S. (2019). Human compatible: Artificial intelligence and the problem of control. Viking.

16. Shapiro, M., Preguiça, N., Baquero, C., & Zawirski, M. (2011). Conflict-free replicated data types. Proceedings of the 13th International Conference on Stabilization, Safety, and Security of Distributed Systems, 386-400.

17. Kwon, J., Kim, T., & Kim, H. (2023). Ensemble methods for large language models: A survey. arXiv preprint arXiv:2310.09542.

18. Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., ... & Christiano, P. (2021). A general language assistant as a laboratory for alignment. arXiv preprint arXiv:2112.00861.

19. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, 214-226.

20. Varia, J. (2010). Architecting for the cloud: Best practices. Amazon Web Services. Retrieved from https://aws.amazon.com/whitepapers/

21. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 3645-3650.

22. Wooldridge, M. (2002). An introduction to multiagent systems. John Wiley & Sons.

23. Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., ... & Barnes, P. (2020). Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 33-44.

24. Patterson, D., Gonzalez, J., Le, Q. V., Liang, C., Munguia, L. M., Rothchild, D., ... & Dean, J. (2021). Carbon emissions and large neural network training. arXiv preprint arXiv:2104.10350.

25. Floridi, L., & Cowls, J. (2019). A unified framework of five principles for AI in society. Harvard Data Science Review, 1(1).