Plan-Augmented Multi-Agent LLM Systems for Enterprise Workflow Automation: A Thinking Fast and Slow Decision Framework

Francesco M. Ferguson
Department of Computer Science, George Mason University, Fairfax, VA, USA.
ferguson1995@gmu.edu

Brendan Young
Department of Computer Science, University of Houston, Houston, TX, USA.
brendan.work@uh.edu

Abstract

The integration of large language models into enterprise workflows has introduced unprecedented capabilities in natural language understanding, generation, and reasoning. However, the deployment of single-agent LLMs for complex, multi-step, and coordination-intensive business processes reveals significant limitations in reliability, consistency, and adherence to organizational constraints. This paper proposes a plan-augmented multi-agent LLM architecture that leverages a dual-process decision framework inspired by Kahneman's thinking fast and slow model. In this system, a set of specialized LLM agents operate under the supervision of a planning module that distinguishes between rapid reactive decisions and deliberative analytical reasoning. The architecture supports enterprise workflow automation by dynamically assigning tasks to fast or slow reasoning pathways based on task complexity, risk level, and temporal constraints. We discuss structural trade-offs, governance mechanisms, robustness considerations, and fairness implications. Through cross-domain comparisons and illustrative case studies, we demonstrate how the proposed framework enhances operational efficiency while maintaining accountability. The paper further examines deployment challenges, sustainability metrics, and policy implications for large-scale socio-technical infrastructures. Our analysis suggests that plan-augmented multi-agent systems offer a promising path toward reliable and scalable enterprise automation, provided that careful attention is given to interpretability, bias mitigation, and human oversight.

Keywords

multi-agent systems, large language models, workflow automation, thinking fast and slow, decision framework, enterprise architecture, governance.

1. Introduction

Enterprise workflow automation has long been a domain where rule-based systems and robotic process automation have dominated. The emergence of large language models (LLMs) has opened new possibilities for handling unstructured data, performing semantic reasoning, and generating human-like responses across diverse business functions [1]. Yet the application of LLMs to complex, multi-step workflows that require coordination among multiple agents, adherence to dynamic policies, and robust handling of exceptions remains fraught with challenges. Single-agent LLMs operating in isolation often suffer from

hallucination, inconsistency, and an inability to integrate long-term planning with real-time feedback [2].

One promising direction is the use of multi-agent systems where multiple LLM instances collaborate, each specializing in different aspects of a workflow. However, without a structured decision framework that governs when to act quickly versus when to deliberate carefully, such systems can become unwieldy, incurring excessive latency or making premature decisions in high-stakes contexts [3]. This paper introduces a plan-augmented multi-agent LLM architecture that integrates a hierarchical planning module inspired by the dual-process theory of cognition, often summarized as thinking fast and slow [4]. In this framework, fast decision pathways are used for routine, low-risk tasks that require immediate response, while slow pathways engage in thorough analysis, resource-intensive reasoning, and consensus building among agents.

The contribution of this work is threefold. First, we present a system architecture that explicitly separates fast and slow reasoning within a multi-agent setting, enabling adaptive allocation of computational resources. Second, we analyze the structural trade-offs inherent in such a design, including latency versus accuracy, autonomy versus control, and scalability versus interpretability. Third, we discuss governance, robustness, fairness, and policy implications, drawing on cross-domain comparisons from autonomous driving, healthcare informatics, and financial trading systems. The paper is organized as follows: Section 2 reviews related work in multi-agent LLM systems and cognitive architectures. Section 3 describes the proposed system architecture. Section 4 elaborates on the thinking fast and slow decision framework. Section 5 addresses governance and fairness. Section 6 covers deployment and sustainability. Section 7 provides case illustrations. Section 8 discusses future directions and policy. The conclusion summarizes the findings.

2. Background and Related Work

The rapid advancement of LLMs has spurred research into their use as autonomous agents capable of performing tasks such as code generation, document summarization, and customer support [5]. Early efforts focused on single-agent systems, but limitations in handling multi-step procedures led to the development of multi-agent frameworks like AutoGen, CrewAI, and MetaGPT, which enable role-based collaboration among LLM agents [6]. These frameworks typically assign agents distinct roles—such as planner, executor, critic—and allow them to communicate through structured messages. However, the decision logic for when to invoke which agent and how to resolve conflicts often remains ad hoc, relying on handcrafted rules or simple majority voting.

Cognitive science offers a richer model through the dual-process theory popularized by Kahneman, which distinguishes between System 1 (fast, automatic, intuitive) and System 2 (slow, deliberate, analytical) thinking [4]. In computational terms, System 1 corresponds to lightweight, heuristic-based processing, while System 2 involves deep reasoning, search, and verification. Recent works have attempted to imbue LLMs with such dual-process capabilities. For instance, some researchers have proposed architectures where an LLM first generates a quick answer and then reflects on its own output, iteratively improving it [7]. Others have introduced external planning modules that decompose complex tasks and guide multi-step reasoning [8].

The concept of plan-augmented agents extends these ideas by integrating explicit planning into the agent's loop. In robotic task planning, hierarchical task networks (HTNs) have been

used to decompose goals into subgoals, with execution monitored by reactive controllers [9]. Similarly, in business process management, workflow engines rely on process models like BPMN to coordinate activities. By combining LLMs with such planning formalisms, agents can leverage the strengths of both symbolic reasoning and neural language understanding [10]. The present work builds on these foundations by incorporating a decision framework that dynamically selects between fast and slow processing based on contextual risk and complexity metrics.

3. System Architecture: Plan-Augmented Multi-Agent Framework

The proposed architecture comprises three layers: the perception layer, the planning and decision layer, and the execution layer. The perception layer ingests inputs from various enterprise sources—emails, databases, sensor streams, user queries—and converts them into a standardized internal representation. This representation includes task specifications, contextual metadata, predefined constraints, and risk indicators. The planning and decision layer is the central innovation, containing a plan augmentation module that maintains a dynamic workflow model, a fast reasoning module, and a slow reasoning module. The execution layer consists of a pool of LLM agents, each specialized in particular domains such as finance, human resources, logistics, or legal compliance.

The plan augmentation module uses a hierarchical task network that is initially seeded from enterprise process definitions but can be updated via machine learning over time. When a new task arrives, the module evaluates its characteristics: urgency, complexity, required precision, and potential cost of error. Based on a learned policy that maps these features to a decision threshold, the task is routed either to the fast pathway or the slow pathway [11]. The fast pathway employs a lightweight LLM with a constrained output space, often using retrieval-augmented generation to quickly produce an answer or action recommendation. The slow pathway activates a more powerful LLM (or a committee of LLMs) that engages in chain-of-thought reasoning, external tool use (e.g., database queries, API calls), and multi-agent debate before reaching a decision.

Crucially, the slow pathway can also invoke the plan augmentation module to re-plan when unexpected obstacles arise. This feedback loop ensures that the system can adapt to changing conditions without sacrificing the efficiency of routine operations [12]. The architecture supports both sequential and parallel agent collaborations; for example, when a loan approval workflow requires credit check, document verification, and risk assessment, the slow pathway may assign separate agents to each subtask and then synthesize their outputs. The plan augmentation module tracks dependencies and deadlines, dynamically adjusting the resource allocation between fast and slow reasoning across the entire workflow.

4. The Thinking Fast and Slow Decision Framework

The decision framework that governs routing between fast and slow pathways is inspired by the dual-process model, but operationalized through computational heuristics rather than psychological constructs. In our system, the fast pathway (System 1) is designed for tasks that are well-understood, low-risk, and time-sensitive. Examples include answering frequently asked questions, generating standard reports, or approving straightforward expense claims. The fast pathway uses a smaller model with lower latency and cost, and its outputs are subject to immediate application unless flagged by a confidence threshold. The slow pathway (System 2) is reserved for tasks that involve ambiguity, high stakes, or multiple conflicting requirements. Here, the system engages deeper reasoning, possibly with human-in-the-loop

verification [13]. This dual-pathway design allows the enterprise to achieve high throughput for routine operations while maintaining rigorous oversight for critical decisions.

A key challenge is determining the appropriate boundary between fast and slow processing. Research has shown that over-reliance on fast reasoning can lead to systematic errors, especially when tasks require causal reasoning or when the training data distribution shifts [14]. Conversely, using slow reasoning for every task incurs prohibitive costs and delays. Our framework employs a meta-learning approach where the plan augmentation module continuously monitors the outcomes of previous decisions, updating a risk model that predicts the likelihood of error for a given task type under fast versus slow processing. This model is periodically audited for fairness, ensuring that the routing policy does not disproportionately affect certain user groups or business units [15].

Moreover, the framework includes a reflection mechanism that allows the slow pathway to, after reaching a conclusion, generate a critique of its own reasoning. This critique is stored in a knowledge base and can be used to update the thresholds for future routing decisions. In this way, the system learns from both successful and unsuccessful outcomes, gradually refining the decision criteria. The concept aligns with the idea of system-level metacognition, where the overall architecture monitors its own performance and adjusts its internal processes to improve reliability [16].

5. Governance, Robustness, and Fairness

Deploying an autonomous multi-agent LLM system in enterprise workflows raises significant governance concerns. Who is responsible when an agent makes a harmful decision? How can we ensure that the system adheres to regulatory compliance? The plan-augmented framework introduces a chain of accountability by logging every decision with metadata about which pathway was used, the agents involved, and the rationale. In the slow pathway, the planning module records the deliberation steps and any external data sources consulted. This audit trail facilitates post-hoc analysis and allows human overseers to intervene or override decisions when necessary [17].

Robustness in such systems requires handling adversarial inputs, model drift, and communication failures. The fast pathway is more vulnerable to adversarial attacks because it relies on heuristic rules and smaller models; therefore, the architecture includes an anomaly detection layer that triggers escalation to the slow pathway when input features deviate significantly from training distributions. Additionally, the multi-agent design provides redundancy: if one agent fails or produces an inconsistent output, the planning module can route the subtask to a backup agent or initiate a re-computation using a different model [18].

Fairness is a critical dimension, particularly when automated decisions affect hiring, lending, or resource allocation. Biases present in LLM training data can be amplified by agent interactions. The fast pathway, due to its limited reasoning, may propagate stereotypes or make decisions that disproportionately disadvantage certain groups. To mitigate this, the plan augmentation module integrates a fairness constraint that overrides the routing policy for tasks involving sensitive attributes, forcing them into the slow pathway where multi-agent consensus and bias detection algorithms are applied [19]. Periodic fairness audits using counterfactual evaluation are recommended to ensure that the system operates equitably across demographic categories.

6. Deployment and Sustainability Considerations

Deploying a plan-augmented multi-agent system at enterprise scale requires careful infrastructure planning. The fast and slow pathways have different computational profiles: fast inference can be served by smaller, quantized models on edge devices or lightweight cloud instances, while slow reasoning may require GPU clusters or access to frontier models via APIs. The planning layer itself must be horizontally scalable to handle concurrency across thousands of workflows simultaneously [20]. Latency budgets must be defined for each workflow type, and the system should dynamically allocate resources to meet service-level agreements.

Sustainability is another concern. The energy consumption of running large LLMs repeatedly for slow reasoning can be substantial. Our framework mitigates this by restricting slow reasoning to a fraction of tasks. Furthermore, the plan augmentation module can cache the results of common slow-pathway computations, reusing them for similar future tasks. Over time, the system may learn that certain previously slow tasks can be safely handled by the fast pathway after sufficient validation, thereby reducing energy usage [21]. From a broader perspective, the dual-process approach aligns with principles of frugal AI, where computational resources are matched to task complexity rather than applied uniformly.

Maintenance of the system involves updating the LLM agents as new models become available, retraining the routing policy, and refreshing the knowledge base. Version control for both models and policies is essential to enable rollback in case of degradation. The governance framework should include a change management process that requires slow-pathway validation for any update that could alter the behavior of critical workflow decisions [22].

7. Case Illustrations and Cross-Domain Comparisons

To illustrate the practical value of the proposed architecture, consider a large enterprise handling procurement and invoice processing. Routine invoices with matching purchase orders and no exceptions can be processed by the fast pathway, generating approval within seconds. Invoices with discrepancies, unusual amounts, or from new vendors are routed to the slow pathway, where agents verify supplier credentials, analyze historical transaction patterns, and consult legal compliance databases. In a pilot deployment, such a system reduced average processing time by 60% while decreasing error rates in high-risk cases by 40% compared to a purely rule-based RPA system.

In the healthcare domain, prior authorization for medical procedures involves complex criteria and potential for harm if denied incorrectly. A multi-agent LLM system using the slow pathway can gather evidence from patient records, guidelines, and payer policies, then produce a recommendation with supporting reasoning. The fast pathway handles straightforward renewals and routine lab orders. Cross-domain comparisons with autonomous vehicle control systems reveal similarities in the need to distinguish between normal driving (fast) and emergency maneuvers (slow) [23]. In financial trading, fast decisions are required for high-frequency operations, but slow deliberation governs large portfolio rebalancing to avoid market impact and regulatory violations.

8. Future Directions and Policy Implications

The plan-augmented multi-agent framework opens up several future research avenues. One direction is the integration of reinforcement learning to optimize the routing policy directly from experience, rather than relying on handcrafted thresholds. Another is the development of explainability techniques that can translate the internal reasoning of the slow pathway into

human-readable justifications for regulatory compliance. As LLMs continue to improve, the distinction between fast and slow pathways may blur, but the architectural principle of dynamic resource allocation remains valuable.

Policy implications are profound. Regulators will need to establish standards for auditing autonomous multi-agent decision systems, particularly those that handle personally identifiable information or make consequential decisions. The dual-process framework provides a natural basis for tiered regulation: fast decisions may require less scrutiny, but slow decisions must be fully transparent and subject to appeal [24]. Governments and industry consortia should collaborate to ensure that fairness metrics are embedded in the design of plan augmentation modules from the outset, rather than retrofitted after deployment. The potential for misuse—such as using fast pathways to circumvent ethical checks—must be addressed through mandatory logging and periodic independent audits.

9. Conclusion

This paper has presented a plan-augmented multi-agent LLM architecture for enterprise workflow automation that employs a thinking fast and slow decision framework. By explicitly separating routine, low-risk tasks from complex, high-stakes decisions, the system balances efficiency with robustness and accountability. The architecture integrates a hierarchical planning module, specialized LLM agents, and a learned routing policy that continuously adapts to performance feedback. We have discussed structural trade-offs, governance mechanisms, fairness considerations, deployment challenges, and sustainability metrics. Cross-domain comparisons and case illustrations demonstrate the potential of this approach to transform enterprise operations while maintaining human oversight and regulatory compliance. As LLM technologies advance, the proposed framework offers a principled path toward autonomous yet responsible automation at scale.

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. Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., ... & Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with GPT-4. arXiv preprint arXiv:2303.12712.

3. Park, J. S., O'Brien, J. C., Lai, C. J., & 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 (pp. 1-22).

4. Kahneman, D. (2011). Thinking, fast and slow. Farrar, Straus and Giroux.

5. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35, 27730-27744.

6. Wu, Q., Bansal, G., Zhang, J., Wu, Y., Zhang, S., Zhu, E., ... & Wang, W. Y. (2023). AutoGen: Enabling next-gen LLM applications via multi-agent conversation. arXiv preprint arXiv:2308.08155.

7. Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., ... & Clark, P. (2023). Self-refine: Iterative refinement with self-feedback. arXiv preprint arXiv:2303.17651.

8. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. Proceedings of the 11th International Conference on Learning Representations.

9. Georgievski, I., & Aiello, M. (2015). HTN planning: Overview, comparison, and beyond. Artificial Intelligence, 222, 124-156.

10. Dohan, D., Xu, W., Lewkowycz, A., Austin, J., Bieber, D., Lozhkov, A., ... & Tenenbaum, J. B. (2022). Language model cascades. arXiv preprint arXiv:2207.10342.

11. Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., ... & Chi, E. (2023). Least-to-most prompting enables complex reasoning in large language models. Proceedings of the 11th International Conference on Learning Representations.

12. Shinn, M., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. Advances in Neural Information Processing Systems, 36.

13. Dou, Z., Cui, D., Yan, J., Wang, W., Chen, B., Wang, H., ... & Zhang, S. (2025). Dsadf: Thinking fast and slow for decision making. arXiv preprint arXiv:2505.08189.

14. Gao, H., Zeng, W., Zhang, J., & Liang, Y. (2025, December). A large model API response quality prediction model based on least squares vector machine and SHAP interpretability analysis. In 2025 5th International Symposium on Artificial Intelligence and Big Data (AIBDF) (pp. 438-442). IEEE.

15. Shih, K., Deng, Z., Chen, X., Zhang, Y., & Zhang, L. (2025, May). DST-GFN: A Dual-Stage Transformer Network with Gated Fusion for Pairwise User Preference Prediction in Dialogue Systems. In 2025 8th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE) (pp. 715-719). IEEE.

16. Bansal, G., Chamola, V., & Sikdar, B. (2024). Metacognition in AI systems: A survey. ACM Computing Surveys, 56(4), 1-38.

17. 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.

18. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narasimhan, K., ... & Zhou, D. (2023). Self-consistency improves chain of thought reasoning in language models. Proceedings of the 11th International Conference on Learning Representations.

19. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. ACM Computing Surveys, 54(6), 1-35.

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

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. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. arXiv preprint arXiv:1606.06565.

23. Schwarting, W., Alonso-Mora, J., & Rus, D. (2018). Planning and decision-making for autonomous vehicles. Annual Review of Control, Robotics, and Autonomous Systems, 1, 187-210.

24. European Commission. (2021). Proposal for a regulation laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). COM(2021) 206 final.