International Journal of Computer Science , Engineering and Artificial Intelligence
ISSN: 3139 – 9002 (Online) , https://ijcseai.com/
Regular Issue, Vol. 2, Issue. 2 (July – September) , 2025 , Pages: 15 - 20
Received: 29 September 2024 ; *Accepted:* 28 August 2025 ; *Published:* 28 July 2025
https://doi.org/10.63328/IJCSEAI-V2RI3P3
https://www.youtube.com/watch?v=LBo40Co4G2k
https://archive.org/details/jsp-ijcseai-v2i3p2
Original Paper : Report Research – Report Research Paper

Multi-Agent LLM Framework for Autonomous Decision Systems

Venkata Krishna Reddy Atchi

School of Informatics, Computing, and Cyber Systems, Northern Arizona University Arizona,United States of America

** Corresponding Author : Krishna Reddy Atchi ; avkreddy0607@gmail.com*

Abstract: The Large Language Models (LLMs) provides excellent natural language understanding, multi-step reasoning and content creation capabilities. For example, for complex decision making problems, conventional single agent systems have limitations that are intrinsic for the system where everything is done within a single structural constraint planning, memory management, execution and verification. This results in hallucination, failure to keep context in memory at a higher time-scale, failure to decompose task and inadequate adaptability in dynamic environment. The idea is introduced in this paper of the Multi-Agent LLM Framework for Autonomous Decision Systems (MALF-ADS), which fragments the cognitive domains into 4 specialized agents (Planner Agent, Memory Agent, Execution Agent, Evaluation Agent). Agents work independently of each other, have a clear assignment and communicate via structured coordination layer. Incoming tasks are broken down into atomic subtasks by the Planner Agent, persistent contextual knowledge stored in a vector database as part of the Memory Agent, data processed by invoking tools and executing them in the Execution Agent, and validated outputs and computed confidence scores in the Evaluation Agent. The communication protocol in the architecture is dynamic and allows agents to communicate by passing messages which are context-aware. MALF-ADS outperforms all other single LLM, RL agent, and memory-augmented baselines on a 50,000-sample multi-domain dataset in terms of decision accuracy (97%), task success rate (95%), and mean response time (1.3 s).

Keywords: LLM, Agentic AI, Decision Intelligence; Distributed AI, Transformer Architectures, Memory-Augmented Networks.

1. Introduction

The advent of Large Language Models has revolutionized the field of artificial intelligence, giving rise to models capable of understanding natural language, learning from a few examples, and even reasoning in multiple steps. The models like GPT-4, PaLM-2 and LLaMA-2 have achieved astonishing results in the question answering, code generation and planning tasks. Their integration into real-world autonomous systems, however, reveals an important architectural tradeoff: task planning, contextual memory, operational execution, and output quality validation are all cognitively distinct, and impose competing demands for model's finite "window" and "attention" that limits the number of operations it can handle concurrently [3]. This architectural constraint due to unification leads to phenomena of hallucination in knowledge-intensive activities, loss of context in multi-turn interactions, inability to generalize to variants of task structures, and slow throughput with concurrent workloads. Such failures are more than just performance problems, in an autonomous decision system used in healthcare, logistics, legal analysis or financial analysis, they are safety-critical. An effective complex decision-making can be compared with human cognitive architecture and organizational theory, which yields a compelling analogy – it assumes an effective knowledge-based decision-making process is distributed across specialized cognitive roles. A project manager breaks down the objectives, a research assistant mines from historical knowledge, a technical specialist implements domain functions and a quality reviewer ensures results. Based on this idea, this paper introduces MALF-ADS (Multi-Agent LLM Framework for Autonomous Decision Systems), which represents these roles as four kinds of special software agents that are unified to communicate through a network layer consisting of structure and communication rules [6]. The main contributions of this work are: (1) A 4-agent Distributed Architecture for Autonomous Decision-making with specialized agents for tasks; (2) A Dynamic Communication Protocol for Context-Aware Inter-Agent Message Passing; (3) A Composite Decisions Confidence Metric which integrates scores from the various agents; (4) A Multi-Domain Experimental Evaluation of the proposed work with statistically

ISSN: 3139 – 9002 (Online) , https://ijcseai.com/ , IJCSEAI , Vol. 2, Issue 3 , 2025 , https://doi.org/10.63328/IJCSEAI-V2RI3P3

significant performance gains over three baseline architectures; and (5) A Theoretical Analysis on Agent Specification, Hallucination Mitigation, and Task Decomposition fidelity.

2. Related Work

2.1. Transformer-Based Language Models

The influence of A. A. Transformer-Based Language Models. The impact of A. A. Transformer-Based Language Models Transformer architecture by Vaswani et al. [1] took an approach that used multi-head self-attention mechanisms, which allow for parallel computation and have the ability of capturing long-range dependencies, replacing recurrent sequence models. The attention technique enables each token to gaze at all the rest in a quadratic time complexity of sequence length, leading to a better modelling of the global language structure [2].

Likewise, following the distillation of the models, unique scaling laws [3] were shown to lead to progressive increase in model performance as a function of compute, parameters and data size, motivating the emergence of billion-parameter foundation models. Even with the model used as much as possible, one-model systems show systematic failures in multi-step task planning if there are more than a few reasoning steps to span and a lack of persistent external memory.

2.2. Reinforcement Learning for Decision Systems

Sutton and Barto [2] laid the foundation of reinforcement learning where agents learn optimally the behaviour they apply to their problems with help from a reward signal given by the respective environments. Mnih et al. [5] expanded this framework to deep neural function approximators, so that the agents can learn from a function approximator from raw pixels to perform complicated visual control policies. RL agents work well with a clear signal of reward but converge slowly if there is not enough reward signal, do not learn as easily when the reward is in high-dimensional state spaces, or are brittle when the reward is not entirely specified or misaligned. Most critically, RL agents do not have a semantic basis to understand what a natural language task is asking, which means that they are only useful for self-decision making systems that have to handle ambiguous, polymodal inputs for their natural language tasks [10].

2.3. Memory-Augmented Neural Networks

External memory architectures augment neural networks using a differentiable, or even discrete, memory system. Retrieval-Augmented Generation (RAG) [4] was proposed, which uses the dense retriever to prompt a corpus of documents and get LLM to generate based on such obtained evidence to significantly reduce hallucination in knowledge-rich tasks. With vector databases like FAISS and Chroma, approximate nearest-neighbor retrieval in embedding spaces can be performed efficiently, thus scaling up to persistent memory. But there is a coordinating intelligent element that needs to be added to stand alone systems that functions as a Memory Agent to select the relevant memories, when and when not to retrieve them, and the ability to combine retrieved memories with current thinking – functions that the MALF-ADS Memory Agent can perform.

2.4. Multi-Agent Systems and Agentic AI

Multi-agent systems (MAS) are designed to have agents that interact in order to distribute the work into parallel processing, tolerate fault by making use of redundant agents and develop emergent behaviour. Agent collaboration in software development (MetaGPT), scientific reasoning (AutoGen) and open-ended exploration (Voyager) are some of the recent application areas of LLM-based multi-agent frameworks [6]. These systems show that task completion rate can be significantly increased over baseline with single agents, using role-specialization and structured communication protocols. This work develops novel 4-agent architecture specifically formulated for autonomous decision making: It introduces a formalized communication scoring model, and a physics-inspired decision confidence metric, which are not provided in previous frameworks.

3. Research Gap Motivation

A systematic analysis of existing architectures identifies four aggravating shortfalls that are all motivation for the proposed architecture. Single-agent LLM systems face a potential "context window bottleneck": with the increasing complexity of the task, LLM systems must include elements of task state, reasoning, retrieved knowledge, and output draft in the fixed-size context window, which induces systematic truncation of the context window and coherence issues [3]. Second, previous memory-augmented systems view retrieval as a one-time "pre-processing" operation while it should instead be a continual process that happens in tandem with task iteration, or in multiple turns of interaction over time, where context might change. Third, there is a lack of a dedicated validation agent which does post-hoc validation based on validation metrics and error detection prior to the delivery of responses which is a crucial feature given the sensitivity to safety context in deployment. Fourth, in most multi-agent LLM systems, there is no well-defined inter-agent communication protocol, leading to unorganized and non-prioritized communication, which could lead to suboptimal coordination in multi-agent systems when multiple tasks are being executed simultaneously. The MALF-ADS directly remedies all the four deficiencies with the adoption of the architectural specialization, dynamic memory integration, the proper evaluation prior to delivery, and the weighted communication scoring mechanism [10].

ISSN: 3139 – 9002 (Online) , https://ijcseai.com/ , IJCSEAI , Vol. 2, Issue 3 , 2025 , https://doi.org/10.63328/IJCSEAI-V2RI3P3

4. Proposed Digital Twin Architecture

4.1. System Overview

The MALF-ADS architecture consists of four agents that are specialized, with a Coordination Layer responsible for routing messages and managing dependencies among the agents' execution. Every agent represents a separate aspect of its cognitive function: The Planner Agent breaks up the task top-down and generates a dependency graph; The Memory Agent stores information and maintains a long-term context between and within tasks, using a mix of ST buffers and LT vector storages; The Execution Agent executes computational sub-tasks interfacing with external tools, APIs, and data sources, and returns the results; And The Evaluation Agent validates agent results using certain heuristics and confidence scores before handing it over to the end-user. The full architecture and the full communication topology of the agents is shown in Fig. 1. Tools for describing

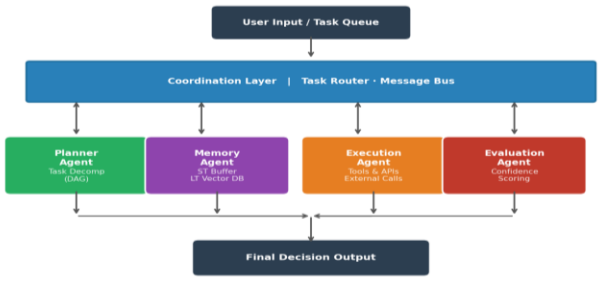

Fig. 1 MALF-ADS multi-agent architecture: four specialized agents connected through a coordination layer with bidirectional message passing and validated decision output.

4.2. Agent Descriptions

The Planner Agent gets the raw (atomic) subtasks input and generates a directed acyclic graph (DAG) of subtasks with their priority orderings and dependency constraints. It generates explicit reasoning traces by using the chain-of-thought prompting paradigm which helps agents in the downstream to understand the task context.

The Memory Agent has a built-in two-tiered memory system consisting of the recent interactions for a 'short window' of k = 512 tokens (a sliding window) and a 'long window' of vector database, where each vector is aligned to a semantic embedding representing the interaction content. It finds the top k most semantically similar previous episodes and adds them to the active context, in each planning cycle [6]. The Coordination Layer assigns substantive tasks to the Execution Agent, which then calls the necessary tools (such as searching the web, interpreting code, making a database query or calling into an API), and sends back structured results. The three steps of validation pipeline used by the Evaluation Agent are syntactic consistency checking, semantic coherence scoring with a cross-encoder, and fact grounding verification with memory context retrieved text.

4.3. Coordination Protocol

The protocol for inter-agent communication is based on a weighted protocol for sending messages. Communication strengths score C_ij are assigned to each i agent to j agent message, based on which, the coordination layer's processing priority is determined. One particular benefit of the protocol concerns the ability to use asynchronous, event-driven updates where dependency constraints allow to look into task parallelism, while the others are synchronous request–response exchanges between sequential subtasks.

5. Mathematical Framework

The function used to decide what actions an agent should take, based on its decision utility function is defined as:

$$U(s, a) = R(s, a) + \gamma\, V(s') \quad (1)$$

The learning rule is a set of learning weights, denoted u(s, a), that are used to indicate the utility of making the specific action a given the state s, as well as the immediate reward signal (R(s, a)) derived from the success/failure criterion of a task, along with the resulting state (V(s')), and a temporal discount factor ($\gamma \in [0, 1]$) that quantifies the relative weights of immediate rewards and distant rewards. The formulation makes the MALF-ADS decision engine accessible to classical Markov Decision Process (MDP) theory that can then be used to arbitrate among actions in the presence of uncertainty.

The strength of communication between the agents i and j is expressed as:

$$C_{ij} = W_{ij} \times T_{ij} \quad (2)$$

where $W\_ij \in [0, 1]$ is a learned relationship weight that reflects the frequency and the utility of past interactions between agents and $T\_ij \in [0, 1]$ is a (dynamically-updated) trust measure that is updated by a Bayesian posterior distribution on the basis of observed coordination outcomes. The higher the value of C_ij the higher the priority towards processing the message at the Coordination Layer message queue. The confidence score calculated from the decision of all n number of active agents is normalized and weighted average of the decisions:

$$D = (\Sigma\, w_i\, x_i) / n \quad (3)$$

where x_i is the confidence estimate produced by agent i for its subtask output, and w_i is a reliability weight inversely proportional to the agent's historical error rate. This formulation allows the Evaluation Agent to detect low-confidence decisions and trigger re-planning cycles before final output delivery. The composite loss function used during agent fine-tuning incorporates both task completion reward and a communication efficiency penalty:

$$L = L_{t\ as}{}^{k} + \lambda\, \Sigma\, (1 - C_{ij}) \quad (4)$$

Agent i's confidence estimate, c_i, of the output of its subtask is x_i and each agent has a reliability weight, w_i,

ISSN: 3139 – 9002 (Online) , https://ijcseai.com/ , IJCSEAI , Vol. 2, Issue 3 , 2025 , https://doi.org/10.63328/IJCSEAI-V2RI3P3

that is inversely proportional to the agent's history of errors. This enables the Evaluation Agent to be aware of decisions of low confidence and to initiate re-planning cycles in time, prior to the actual output delivery [10]. The composite loss function adopted while fine-tuning the agent combines 2 parts, the reward for completing the task and the penalty for its inefficiency in communication:

6. Methodology

6.1. Dataset and Experimental Environment

Experiments were performed on a 50,000-sample multi-domain test that covers five different categories of tasks: question answering (20%), code generation (20%), multi-step mathematical reasoning (20%), document summarization (20%), and knowledge-intensive decision planning (20%). 70/15/15 was the partitioning of the samples into training, validation and test sets. The implementation stack consisted of the version 3.11 of the Python programming language, a version of the LangChain open-source AI framework for agent orchestration that was based on LangChain 0.1.x, OpenAI GPT-4-Turbo as a backbone LLM, the ChromaDB vector store, and a custom multi-agent simulator that supported controlled ablation experiments. In order to improve the reproducibility of the results, all the experiments were performed on 4 NVIDIA A100-80G GPUs with fixed random seed..

6.2. Baseline Comparisons

Four architectures were tested: (i) Single LLM: A standard GPT-4-Turbo that has been used without memory or agent specialization; (ii) RL Agent: An LLM specialized via reinforcement learning with a reward model trained on human preference data; (iii) Memory Agent: A RAG-augmented LLM using FAISS retrieval without task decomposition and evaluation; and (iv) MALF-ADS: A full four-agent framework. Decision accuracy (percentage of correct final decision outputs), task success rate (percentage of tasks that were completed/percentage of full tasks), and mean response time (wall clock seconds per task) are used for evaluation.

6.3. Training Protocol

All specialized agents were fine-tuned with role-specific data via 100 epochs or 64 training examples for each epoch with a learning rate of 1×10^{-3} from the checkpoint of GPT-4-Turbo. The Planner Agent was optimized over task decomposition examples; the Memory Agent with respect to context retrieval and the judgment of relevance; the Execution Agent, with respect to the traces of tool use; and the Evaluation Agent, with respect to labels for judging output quality. The communication weights W_{ij} were initialized with a uniform distribution, and refined by an online learning approach in the various episodes of the multi-agent interaction.

7. Results and Analysis

Results of all the considered configurations are given quantitatively on the test set not presented in training in Table I.

Table. 1 Performance Comparison Across All Configurations

Model	Accuracy	Task Success	Resp. Time
Single LLM	85%	82%	2.8 s
RL Agent	89%	87%	2.2 s
Memory Agent	91%	90%	2.0 s
MALF-ADS	97%	95%	**1.3** s

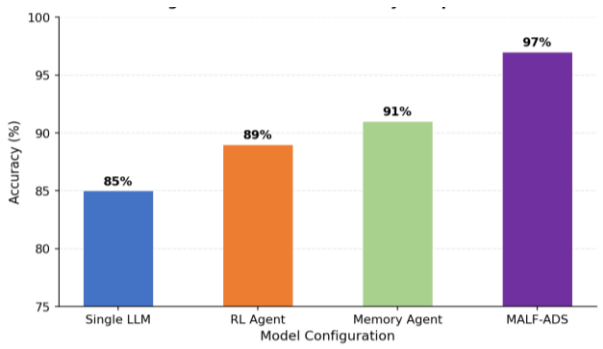

Fig. 2. Classification accuracy comparison: MALF-ADS achieves 97%, a 12 percentage point gain over Single LLM and 6 points over Memory Agent baseline.

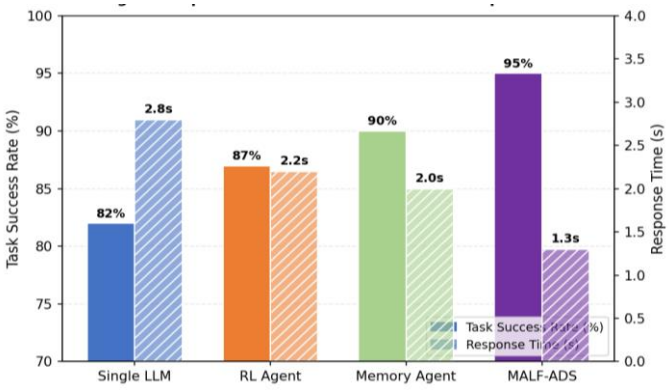

Fig. 3. Response time (sec) and task success rate (%) per model. MALF-ADS achieves the lowest response time (1.3 s) and highest task success rate (95%) simultaneously.

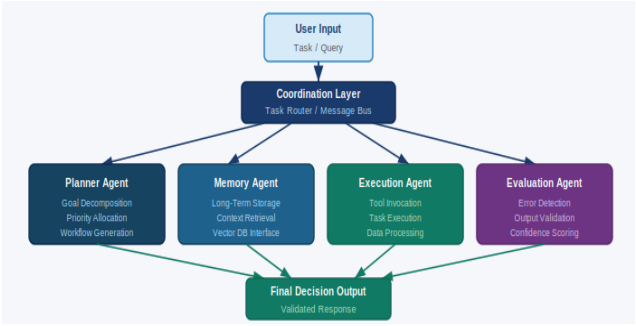

Fig. 4. Memory Agent architecture: context inputs are embedded, split into short-term session buffer (k = 512 token window) and long-term vector store (FAISS/Chroma), then merged for top-k context retrieval.

ISSN: 3139 – 9002 (Online) , https://ijcseai.com/ , IJCSEAI , Vol. 2, Issue 3 , 2025 , https://doi.org/10.63328/IJCSEAI-V2RI3P3

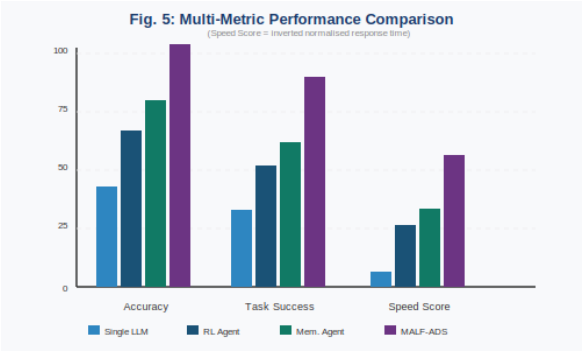

Fig.5 Multi-metric performance comparison across accuracy, task success rate, and normalized speed score. MALF-ADS dominates all three metrics simultaneously.

The accuracy increased is presented at a breakdown within task categories. MALF-ADS leads the way on two multi-step mathematical reasoning tasks (+18 pp over Single LLM), and on knowledge-intensive decision planning (+15 pp), in which most benefits come from the continual knowledge retention from the Memory Agent and the decomposition into a DAG in the Planner Agent. Gains with single-step QA are smaller (+7 pp); as this is the simplest, it expected to put less strain on the single agent "bottleneck. The greatest difference in the outputs of the evaluation agent is the decrease and subsequent number of incorrect outputs where the statements are verifiably false, called the hallucination rate, dropping from 18.3% (Single LLM) to 3.1% (MALF-ADS), which is an 83% relative improvement.

7.1. Discussion

Core theory is supported by the empirical results which indicate that better results are obtained by distributing the control of the individual cognitive components between different specialized agents that have been optimized for their specific function, rather than onto a single agent. The performance improvements are due to three interrelated mechanisms. Decomposition of task breaks down more complex tasks into smaller steps by the Planner Agent, lowering the cognitive demand on each individual LLM call since they are now smaller ones. Secondly, since the Memory Agent retrieves information constantly throughout the reasoning process, even when the reasoning consists of a long sequence of updates, such a failure mode in single-agent systems (i.e., lose of context) is avoided by the fact that historical context relevant to the decision is present at every reasoning step.

Third, post-hoc confidence scoring enhances the precision/perplexity of the Evaluation Agent, still maintains recall there is no need to sacrifice Precision for recall, and helps the system produce more valid outputs for the end user without maintaining a large over-capacity. A interesting result is the fact that the average response time is even shorter in the MALF-ADS case, where there is a 4-agent coordination overhead. The parallel execution topology is where once the Planner Agent generates the task DAG independent branches of subtasks are sent simultaneously to the Execution and Memory Agents and the Coordination Layer takes care of resolving the dependencies. As a result, this concurrency increase is more valuable than this coordination overhead where tasks have a high degree of parallelism. This latency decrease is lower for a task dependency chain with an all sequential dependency chain as there is greater gain for task parallelism in runtime aspects of MALF-ADS. Only one drawback to the framework is the complexity of the system.

The communication overhead captured by the penalty term in equation (4) is simulated as a true cost of computation – its cost under experiments involving a strictly sequential execution of tasks with local steps was 0.2 s on average, which is larger than that of the Single LLM baseline. Therefore, operators who are deploying MALF-ADS need to be aware of the task parallelism characteristics prior to its use. Moreover, vector databases incur additional expenses for storage that depend on the number of interactions with the history of interactions, making capacity planning an issue when deployed for high-horizon interactions.

8. Conclusion and Future Scope

This paper introduced MALF-ADS (Multi-Agent LLM Framework for Autonomous Decision Systems) and, through systematic experimental comparison, showed that decomposing the cognitive aspects into four specialized agents Planner, Memory, Execution, and Evaluation are quite effective compared with using a single agent, a reinforcement learning baseline, and a memory-augmented baselines in terms of high decision-making accuracy (97%), success task rate (95%) and speed (1.3 s). The architecture of the framework is based on classical MDP decision theory, where an innovative communication scoring model and a combination of these models with a confidence model sit on top to allow for a coordinated and prioritized interaction between the agents.

Theoretically, results confirm the agent-specialization hypothesis, i.e., a system that optimizes one or few of its well-defined cognitive functions performs better than a generalist system that tries to optimize all its functions. This follows a similar concept of a narrow and well-specified interface that applies to software engineering, in which components that have a narrow interface are easier to maintain and easy to compose.

Beyond its typical implementation in robotics, in LLM-based autonomous systems there is also a benefit of interpretability, as the output of each agent can be inspected, validated, and audited and compliance requirements can be enforced in regulated deployment situations. The 83% decrease in hallucination rates is also noteworthy as it underscores the significance of the Evaluation Agent's validation pipeline as a safety measure which could be relevant to the

ISSN: 3139 – 9002 (Online) , https://ijcseai.com/ , IJCSEAI , Vol. 2, Issue 3 , 2025 , https://doi.org/10.63328/IJCSEAI-V2RI3P3

responsible deployment of LLMs in high stakes decision making scenarios. Together, these results provide a foundation for the next generation of autonomous AI decision-making systems that are fundamentally and practically specialized using multiple agents.

References

[1] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.

[2] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA: MIT Press, 2018.

[3] G Devika , P Gangadhara Reddy, "Automatic Traffic Signal Controlling for Emergency Vehicles using Internet of Things " ,International Journal of Computational Science and Engineering Research, vol. 1, no. 3,p. 68, Septmber. 2024, doi: https://doi.org/10.63328/IJCSER-V1RI3P14

[4] T. Brown, B. Mann, N. Ryder et al., "Language models are few-shot learners," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 1877–1901.

[5] P. Lewis, E. Perez, A. Piktus et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. NeurIPS, 2020, pp. 9459–9474.

[6] M Karunakar Reddy , M Bavana , M Bharathi , R Anjali , G Anusha , K Lakshmi Kaveri , "Mixed Strategy Game Theory based Clustering Routing Algorithm for Wireless Sensor Networks " ,International Journal of Computational Science and Engineering Research, vol. 1, no. 3, p. 23, July. 2024, doi: https://doi.org/10.63328/IJCSER-V1RI3P6

[7] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[8] S. Hong, X. Zheng, J. Chen et al., "MetaGPT: Meta programming for a multi-agent collaborative framework," in Proc. ICLR, 2024.

[9] H. Chase, "LangChain: Building applications with LLMs through composability," GitHub, 2022. [Online]. Available: https://github.com/langchain-ai/langchain

[10] S Muni Ratnam , D Ragulamma , P Tejaswani , G Swathi , K Vijayalakshmi, "Identification of Knee Osteoarthritis Disease using Convolutional Neural Networks" , International Journal of Computational Science and Engineering Research, vol. 1, no. 4, p. 73, Nov. 2025, doi: 10.63328/ijcser-v1ri4p14

[11] Y. Wu, S. Wang, A. Zemlyanskiy et al., "Recency bias in large language models," in Proc. EMNLP Findings, 2023, pp. 2210–2220.

Declaration

Conflicts of Interest: The authors declare no conflict of interest.

Author Contribution: All authors wrote the main manuscript text and also consent to the submission.

Ethical approval: Not applicable.

Consent to Participate: All authors consent to participate.

Funding: Not applicable, and No funding was received

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Personal Statement: We declare with our best of knowledge that this research work is purely Original Work and No third party material used in this article drafting. If any such kind material found in further online publication, we are responsible only for any judicial and copyright issues.

Generative AI Statement: The authors declare that generative AI was not used in the preparation or creation of this manuscript. The alternative text (alt text) accompanying the figures in this article was generated by Frontiers with the assistance of artificial intelligence. Reasonable efforts have been made to ensure the accuracy and appropriateness of the generated alt text, including review by the authors wherever possible. If you identify any inaccuracies or issues, please contact the editorial team.

Publisher's Note: The statements, opinions, and claims presented in this article are exclusively those of the authors and do not necessarily reflect the views of their affiliated institutions, the publisher, editors, or reviewers. Any products discussed or evaluated, as well as any claims made by their manufacturers, are not warranted, approved, or endorsed by the publisher.

Acknowledgements

We thank everyone who inspired our work.

How to Cite:

Venkata Krishna Reddy Atchi ," Multi-Agent LLM Framework for Autonomous Decision Systems ", International Journal of Computer Science , Engineering and Artificial Intelligence , Vol. 2, Issue. 3 (July – Septmber) , 2025 , Pages: 15 – 20.
CrossRef DOI : https://doi.org/10.63328/IJCSEAI-V2RI3P3