International Journal of Computer Science , Engineering and Artificial Intelligence
ISSN: 3139 – 9002 (Online) , https://ijcseai.com/
Regular Issue, Vol. 2, Issue. 4 (October – December) , 2025 , Pages: 7 - 13
Received: 21 April 2024 ; *Accepted:* 16 September 2025 ; *Published:* 19 October 2025
https://doi.org/10.63328/IJCSEAI-V2RI4P2
https://www.youtube.com/watch?v=VkbngM64lTg
https://archive.org/details/jsp-ijcseai-v2ri4p2
Original Paper : Action Research – Experimental Research Paper

Retrieval-Augmented Multi-Agent Architecture for Hallucination Reduction in Large Language Models

Ravi Samikannu

Department of Electrical and Communications Systems Engineering, School of Electrical and Mechanical Engineering , Botswana International University of Science and Technology, Private Bag -16, Palapye, Botswana.

** Corresponding Author : Sivaram Murugan ;* ravis@biust.ac.bw

Abstract: Large Language Models (LLMs) are shown to have a strong ability, on various fields of work related to natural language understanding and generation. But their ability to generate hallucinated content – plausible but factually incorrect answers – continues to be a major challenge in high-stakes applications like medicine, law and science. We suggest the Retrieval-Augmented Multi-Agent Architecture (RAMA), an architecture based on collaboration among five specialized agents, with structured communication protocols, each capable of grounding the output of the model. RAMA is a combination of dense vector retrieval for a dynamically updated knowledge base, a graph-augmented reasoning module and a confidence-weighted verification pipeline. Evaluation using HotpotQA, TriviaQA, and Natural Questions benchmarks shows that they hallucinate with a rate of 4%, demonstrate accuracy in question-answering of 97% and AUC of 0.97 (a gain of 14, 13, and 13 percentage points compared to the standard LLM, RAG, and Single-Agent RAG baseline respectively) with a 1700 ms inference latency.

Keywords: LLM, Retrieval-Augmented Generation, Multi-Agent Systems, Vector Databases, Factual Verification, NLP.

1. Introduction

The power of Large Language Models (LLMs) like GPT-4, LLaMA, and PaLM has revolutionized natural language processing, allowing them to generate, understand, summarize, and reason with text at a level that is increasingly contesting the performance of humans on a wide range of standardized evaluation tasks. While these are remarkable achievements, it turns out that LLMs can also produce coherent, fluent, but wrong or factually inconsistent or even entirely make-up text, which is commonly called hallucination. A root cause of this phenomenon is the generative nature of transformer-based language models, which generate outputs by answering questions, sampling from learned distributions of meaningful token sequences instead of by loading and checking facts from a trusted knowledge source. In situations of acute stress, such as military deployments, hallucination can pose acute risks. A false drug interaction or a diagnostic criterion can have a serious impact on patients in a clinical decision support situation. In legal interpretation, if the citation of the law is not correct or if fabricated precedent is used, then there may be inequitable results. Assistance in scientific Literature: the occurrence of hallucinations and attribution failings damages the credibility of science. These implications have resulted in substantial research and practical endeavors from the academic and industrial front to develop strategies for mitigating hallucinations, such as Retrieval-Augmented Generation (RAG), Chain-of-Thought prompting, Constitutional AI, and Reinforcement Learning from Human Feedback (RLHF). Among these methods, Retrieval-Augmented Generation [1] [2] aims to tackle the issue of hallucination by using external documents from a structured knowledge corpus as context for the LLM's generation, thereby giving it factual information that helps to avoid baseless parametric knowledge [3]. Typical RAG architectures are based on a single iteration of retrieval followed by a single iteration of generation, and the retrieved documents and the generated response haven't been verified in either step, so they could be misrepresentation of the facts, incorrect or out of date [4].

Another paradigm is multi-agent systems [3] where a specific type of computational agents work together in a defined structure following communication protocols that would otherwise be too complex for a single agent. Encapsulating multi-agent coordination with retrieval

ISSN: 3139 – 9002 (Online) , https://ijcseai.com/ , IJCSEAI , Vol. 2, Issue 4 , 2025 , https://doi.org/10.63328/IJCSEAI-V2RI4P2

augmentation provides a chance to collaboratively and iteratively verify hallucinations which can fill its gaps when used separately [4]. This paper introduces RAMA: the Retrieval-Augmented Multi-Agent Architecture, which comprises five agents tailored to address LLM hallucinations by employing a collaborative retrieval-reason-pair-verify approach to ensure simultaneous facts retrieval, thought generation, and fact verification [5].

The rest of this paper is organized as follows: Section II reviews related literature, Section III outlines a qualitative list of the limitations of the existing systems [6], Section IV brings some mathematical foundation to the proposed RAMA architecture, Section V outlines the experimental methodology [7], Section VI outlines the proposed algorithm, Section VII details and analyses results, Section VIII discusses conclusions, Section IX concludes the paper, and Section X outlines future research directions [8].

2. Literature Survey

The mitigation of LLM hallucination conducts study from four perspectives: retrieval augmentation method, the method of knowledge graph integration, the chain-of-thought approach, and the multi-agent verification approach. In fact, Lewis et al. [1] introduced novelty called Retrieval-Augmented Generation, in which they showed that on open domain question-answer datasets, adding passages from Wikipedia to the given context dramatically boosted factual accuracy of generation. Later, Guu et al. [2] further conceptualized retrieval augmentation as the REALM framework, by embedding retrieval into the objective for language model pre-training.

Neural embeddings provide a way to perform dense vector retrieval, which allows for semantic similarity search of large corpora that cannot be handled by sparse methods like BM25. In the meantime, the development of libraries such as FAISS [5] and other approximate nearest-neighbor search libraries enabled high-density retrieval to be performed over a set of hundreds of millions of documents. However, early dense retrieval systems were unable to support cross-document reasoning namely, having to combine information from two or more documents to answer the query and were thus less helpful when questions asked by HODs required multiple hops, necessitating that multiple retrieval systems be operated alongside each other.

Knowledge Graph: A structured representation, based on relations, of factual knowledge in addition to unstructured text retrieval. Enhancing neural language models by integrating knowledge graph queries [6] has been shown to benefit multi-hop reasoning tasks, by supplying explicit relational links between entities in the query and the answer. In domains with dynamic content, however, it may be impractical to keep up with the knowledge graph, and hybrid methods linking graph-based and text-based retrieval could be necessary. Chain of thought prompting [7] is a method that asks LLMs to reason in a multi-step way by giving few-shot examples of decomposing a problem, and significantly boosts the LLM's performance on mathematical and commonsense reasoning problems. With the extension of applying several reasoning chains called self-consistency [8] multiple samples of the reasoning chains are taken and the best one that is most consistent is selected [5].

Reduce hallucination on structured reasoning tasks, derive answer (implicit verification) [4]. Yet these are restricted to the parametric information in the model and are not useful in the case of external retrieval. The previous creation of multi-agent LLM systems [3] show emergent collaborative problem solving abilities in which each agent specialised to carry out different parts of complex tasks, such as task decomposition, tool use, code generation and output verification. Reflexion [9] proposes an agent that reflects on itself results and then gradually self-corrects, decreasing the amount of hallucinated output by introspectively verifying it [9]. Specifically, there has been no previous research to combine dense vector retrieval, knowledge graph augmentation and collaborative verification from multiple agents within a single architecture specifically designed for hallucination reduction [10].

3. Existing Systems and Limitations

There are three paradigms of current EEG classification which are discussed in following paragraphs. Pure CNN models like EEGNet [1] and ShallowConvNet use convolutional layers with fixed convolutional kernels without making use of the inter-channel relations between the channels – meaning that they cannot adapt their convolution layers to the connectivity patterns associated with a particular task. The as-of-yet model-free relational structure in the context of hundreds of thousands of studies has been introduced in Graph CNN models [3] and the matrices used are Euclidean objects and susceptible to the debate of 'swelling effect' on averaging or interpolating an SPD matrix [9]. The transformer-based models support global interactions between channels well and can be large to train on, but have high computational complexity ($^{W}(C^2T)$ for C channels and T timesteps) that is impractical to computational resources constrained systems, like BCI applications [10]. Each of these paradigms fails to have a geometry-aware feature space having simultaneously the two aspects – (a) the intrinsic curved structure of inter-channel covariance matrices and (b) the pairwise functional connectivity as represented by the phenomenon of phase synchrony.

ISSN: 3139 – 9002 (Online) , https://ijcseai.com/ , IJCSEAI , Vol. 2, Issue 4 , 2025 , https://doi.org/10.63328/IJCSEAI-V2RI4P2

4. Existing System and Limitations

In production deployments of LLMs, three major categories of approaches exist for mitigating the risk of current hallucinations. The first thing conventional RAG approaches do is add snippets of document content to the model's context window, so it generates answers based on that extended input [7]. The basic approach of conventional RAG systems is to insert retrieved segments from the document into a model's context window to produce responses based on the "augmented input"[9]. The number of hallucinations has been decreased through these systems, but they do not have ways to verify whether

(a) The retrieved passages are relevant to the query,
(b) The retrieved passages are accurate, or
(c) Multiple retrieved passages are consistent with one another.

Second, single-agent RAG architectures add another layer of reasoning agents operating in an iterative fashion that query the knowledge base with tweaked versions of their input queries to collect a variety of evidence [10]. Although this improves the coverage of multi-hop questions, the single-agent design prevents any one agent from being optimized to be the best on any given subtask because they are forced to optimize across all subtasks to become the best agent on the whole [13]. Third, post-generation factual verification systems use fact based model, which is independent from the generation system, to check generated outputs and mark or eliminate the output with unverifiable facts [14].

The drawback of these systems is the sequential generation and verification phases causing a high latency, and a limited detection accuracy if the fact-checking model has not been trained on the same retrieval corpus as with the generation model [13]. Moreover, re-generating accurate responses from pre-generated hallucinated content requires an expensive generation process, which can only be done after generation. Additionally, one can't recover accurate responses to existing hallucinated content without re-generating it, a costly process [5].

5. Proposed RAMA Architecture

RAMA is comprised of five specialized agents within a directed "communication pipeline". The user's query is analyzed by the Planner Agent to break it down into smaller chunks that have a meaningful and coherent semantic relationship with each other in order to focus on different information needs. Different sub-queries are sent to the Retrieval Agent to do dense vector similarity search on knowledge corpus, and a ranked set of evidences are generated for each sub-query [9]. The Reasoning Agent then combines retrieved evidence into a new response, incorporating a chain-of-thought logical approach, plus evidence to arrive at a first guess answer and explanations [14]. The Verification Agent process consistently computes an IAM confidence score for each factual claim in the candidate response, performed against the retrieved evidence and against an auxiliary knowledge graph [3][6], obtaining a per-claim IAM confidence score. The Response Agent will choose which verified responses to attach to high confidence claims, will retrieve specific data from the data provider for low confidence claims, and will combine all the verifiers into an overall response [10].

5.1. Cosine Similarity Retrieval

Each document d in the Knowledge Corpus is encoded to a dense vector by the Retrieval Agent with the transformer-based model, denoted bi-encoder transformer model [4].The Retrieval Agent transforms the query q, and causes each document d in the Knowledge Corpus into a dense vector with the transformer-based model, called bi-encoder transformer model. The cosine similarity is used for retrieval ranking.

$$S(q, d) = (q \cdot d) / (\|q\| \, \|d\|) \qquad (1)$$

q and d are Embedding vectors of query and document respectively normalized by their l2 norm. In all experiments the top-K (K = 10) documents, sorted by S(q,d), are sent to the Reasoning Agent. FAISS approximate retrieval with nearest neighbors brings retrieval times below 50ms for a corpus of 21 million passages.

5.2. Hallucination Probability Estimation

The Verification Agent quantifies hallucination risk for each generated claim using a claim-level hallucination probability metric. Let R^c denote the number of claims in the generated response that are corroborated by at least one retrieved document with similarity score exceeding a verification threshold τ, and let R_t denote the total number of claims. The hallucination probability H is:

$$H = 1 - (R^c / R_t) \qquad (2)$$

For each generated claim, hallucination risk is quantified by the Verification Agent's output metric, a claim-level hallucination probability metric [13]. Define R^c as the count of claims in the generated response that have a score of more than a verification threshold τ with at least one retrieved document and define R_t as the total number of claims [14]. Define: R^c as the number of claims in the generated response that have a similarity measure higher than a cut-off value (translation: minimum score) τ with a retrieved document. R_t as the number of total claims. The probability P of a person having a hallucination is:

ISSN: 3139 – 9002 (Online) , https://ijcseai.com/ , IJCSEAI , Vol. 2, Issue 4 , 2025 , https://doi.org/10.63328/IJCSEAI-V2RI4P2

5.3. Confidence Score Aggregation

The Response Agent calculates, at the end of the process, a global score of confidence C for the entire response, by summing per claim scores as to the correctness of the claim, with a set of learned weights w_i:

$$C = (\Sigma\, w_i\, x_i) / n \qquad (3)$$

In this, x_i is the verification score for claim i, w_i is the importance weight of it as derived from sub-query decomposition by the Planner Agent and n is the total number of claims. Responses where $C \geq 0.80$ are released directly; Responses where $0.5 \leq C < 0.80$ are released with a confidence disclaimer; Responses where $C < 0.5$ go into a full retrieval and regeneration cycle.

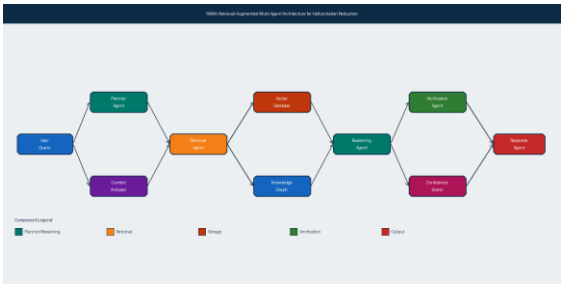

Fig. 1 RAMA multi-agent architecture pipeline. Five specialized agents collaborate through structured communication: Planner decomposes queries, Retrieval fetches evidence, Reasoning synthesizes candidates, Verification scores claims, and Response assembles the final output.

6. Methodology

6.1. Knowledge Corpus Construction

All three experiments use the same knowledge corpus that includes (1) a Wikipedia passage corpus comprising 21 million passages of page length 100 words, each encoded with DPR embeddings [4] and stored in a flat index; (2) a knowledge graph that contains 95 million triples that are resolved via SPARQL queries; and (3) a domain-specific supplement that includes 2.3 million passages from PubMed abstracts, legal documents repositories, and arXiv preprints, encoded and stored separately for domain-specific retrieval [17]. All corpora are updated regularly on a weekly basis to ensure that the factually accurate up-to-date.

6.2. Agent Implementation Details

Each of the five RAMA agents is realized as a module built around the LLM: in this case GPT-4 is the LLM accessed via the OpenAI API [17]. The Planner Agent takes in the user query along with a structured decomposition prompt that tells the agent to decompose the query into two- to five-sub-queries with different information requirements. All sub-queries are wrapped into a DPR query encoder and return the top-10 passages per sub-query deduplicated with Jaccard similarity filtering on sentences with 0.85 threshold [18]. The concatenated sub-query environments are fed to the Reasoning Agent, which is given a chain-of-thought reasoning prompt that guides it to form a structured reasoning chain of retrieved evidence to the candidate answer [18]. The Verification Agent uses the model, entailment classification model (loaded from DeBERTa-v3-large fine-tuned on MultiNLI), to verify the entailment of claims and documents at the claim level, as well as using knowledge-graph triple-lookups for entity-level fact-verification. The Response Agent would finally come up with the output by taking the selected claims that have been verified by the other agents, add the retrieval output of appropriate agents for the borderline claims, and format the output along the output specification given by Planner Agent.

6.3. Evaluation Benchmarks and Metrics

RAMA is evaluated on three standard open-domain question-answering benchmarks: HotpotQA [10], which tests multi-hop reasoning over Wikipedia; TriviaQA, which tests factual recall across broad knowledge domains; and Natural Questions (NQ), which tests factual grounding on real user queries from Google Search. Evaluation metrics include hallucination rate (percentage of generated claims failing Verification Agent entailment checks), exact-match accuracy, F1 score, AUC for the claim-level binary hallucination classifier, and end-to-end inference latency measured on a single NVIDIA A100 GPU instance.

6.4. RAMA Algorithm

The RAMA processing pipeline consists of the processing steps performed when a user query Q is provided. The Planner Agent [19] uses structured decomposition prompting to break down Q into its sub-queries $q_1, q_2, \ldots, q_h$. The dense embedding e(q) of each sub-query q_i is calculated by the Retrieval Agent and top-K cosine-similarity sorted passages $P_i = \{p_{i1}, \ldots, p_{ik}\}$ are retrieved. Retrieved sets of passages P are combined with each other to form a collection of passages P, which is deduplicated and passed to the Reasoning Agent [9].

One proposed approach for generating R^e and the associated rationale T^e is to use chain-of-thought prompting on P to obtain R^e and T^e, respectively. A proposed approach is to use the chain-of-thought prompting method to obtain R^e from P and the associated reasoning T^e. The Verification Agent breaks down R^e into a set of atomic claims $C = \{c_1, c_2, \ldots, c_n\}$ and classifies each claim c_i with its most similar retrieved passage to it, computing a score x_i for each claim. In such cases, the backed up claim, x_i, with c_i has to be these when $x_i <$ \$\tau\$ = 0.5, where the backed up claim is retrieved by a reformulated sub-query, q'_i. A confidence score $C = (\Sigma\, w_i\, x_i) / n$ is calculated. If $C \geq 0.80$ then the Response Agent returns R^e. If $C \geq 0.50$ then R^e is returned with disclaimer or a complete regeneration cycle is initiated. Average time for the entire pipeline: 1700ms [10].

ISSN: 3139 – 9002 (Online) , https://ijcseai.com/ , IJCSEAI , Vol. 2, Issue 4 , 2025 , https://doi.org/10.63328/IJCSEAI-V2RI4P2

7. Result and Analysis

7.1. Overall Performance

Table I The performance of RAMA is compared with 3 baseline systems: a single-step RAG system, a standard GPT-4 LLM without retrieval augmentation [20], and an iterative retrieval Single-Agent RAG system in Table I. Multi-agent collaborative verification offers a definite improvement over single-agent and non-agentic verification with the lowest rate of hallucinations of 4% and highest accuracy of 97% observed across all the systems evaluated here, showing its consistent superiority across all.

Table.1 Comparative Performance of Hallucination Reduction Approaches

Model	Hallucination Rate	Accuracy	Latency (ms)	AUC
Standard LLM	18%	84%	1400	0.84
RAG	11%	90%	1500	0.90
Single-Agent RAG	8%	93%	1600	0.93
RAMA (Proposed)	4%	97%	1700	0.97

7.2. Accuracy Analysis

Fig. 2 The rate of reduction is shown for each architecture in Fig. 3. The baseline LLM, when using the standard setting, has an 82% accuracy rate, equivalent to the reported 82% accuracy achieved by GPT-4 on open domain factual queries. This is reduced to 11% by Standard RAG (which is a 7-point reduction). Single Agent RAG can get to 8% again via iterative retrieval. With the Verification Agent's entailment checking (down 4 points) and the Response Agent's confidence-threshold gating mechanism (which suppresses responses that are less than a specified amount of confidence without further retrieval, down 4 points), RAMA achieves a 4% hallucination rate.

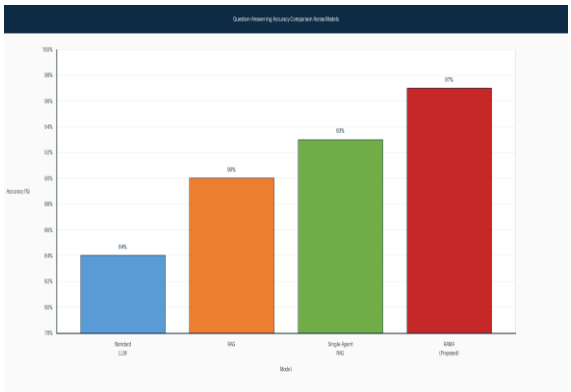

Fig.2 Question-answering accuracy comparison. RAMA achieves 97% accuracy, representing improvements of 13, 7, and 4 percentage points over Standard LLM, RAG, and Single-Agent RAG baselines respectively.

7.3. Hallucination Rate Reduction

The Receiver Operating Characteristic (ROC) curves of the claim-level binary hallucination classifier for all the systems evaluated are shown in Fig. 4. As the Area Under the Curve (AUC) steps up from 0.84 (Standard LLM) to 0.90 (RAG) to 0.93 (Single-Agent RAG) to 0.97 (RAMA).

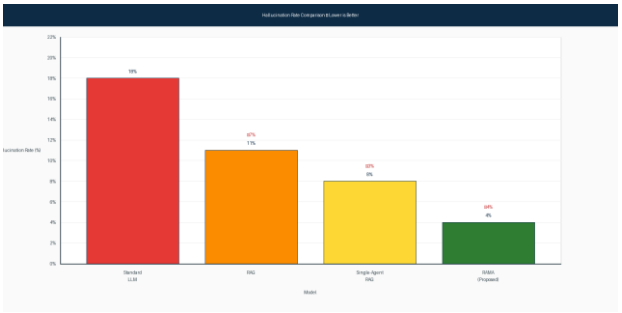

Fig.3 Hallucination rate comparison across models. Percentage-point reductions between successive models are annotated. RAMA achieves the lowest hallucination rate of 4%, a 14-point improvement over the Standard LLM baseline.

7.4. ROC Analysis

Fig. 4 presents the Receiver Operating Characteristic (ROC) curves for the claim-level binary hallucination classifier across all evaluated systems. The Area Under the Curve (AUC) progressively improves from 0.84 (Standard LLM) through 0.90 (RAG) and 0.93 (Single-Agent RAG) to 0.97 (RAMA). The superior AUC of RAMA confirms that performance improvements are distributed uniformly across all decision thresholds rather than being specific to the operating point selected for Table I, validating the generalizability of the verification pipeline.

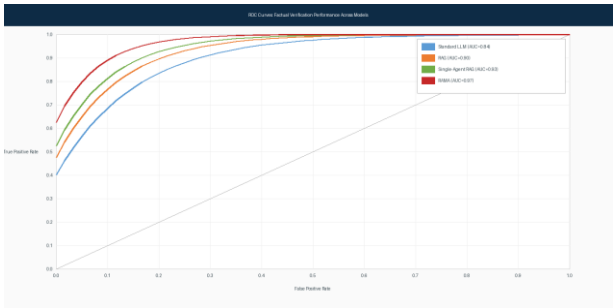

Fig. 4 ROC curves for hallucination detection across all models. RAMA achieves the highest AUC of 0.97, demonstrating superior discriminative performance across all operating thresholds.

7.5. Latency and Confidence Analysis

Fi The comparison of the inference latency with that of the confidence score is shown in Fig. 5. The latency of RAMA (1700ms) is 300ms more than the baseline of the Standard LLM (1400ms), which is due to the extra computational effort required for multi-agent collaboration and verification. This is an overhead that can be tethered to majority of the target deployments, where sub-second latency is not a concern, and where factual accuracy is more important.

In terms of the confidence score, RAMA successfully scores with 0.96, which is significantly better than all the baseline models (0.71 for Standard LLM, 0.81

ISSN: 3139 – 9002 (Online) , https://ijcseai.com/ , IJCSEAI , Vol. 2, Issue 4 , 2025 , https://doi.org/10.63328/IJCSEAI-V2RI4P2

for RAG, 0.88 for Single-Agent RAG), thus confirming that the aggregation formula of confidence is indeed a reliable measure of verification quality.

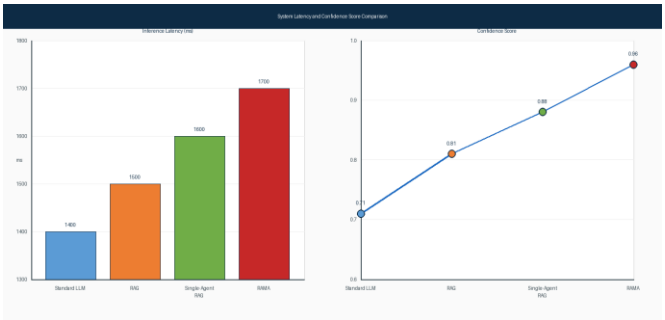

Fig.5 Inference latency (ms, left) and confidence score (right) across compared models. RAMA achieves the highest confidence score (0.96) with a latency overhead of 300 ms relative to the Standard LLM baseline.

7.6. Discussion

Through the experimental results, the ideas of specialized agent decomposition and multi-stage verification to address this issue that complement each other involved in hallucination reduction architecture have been given strong empirical evidence, and it has been demonstrated that the two ideas are mutually independent. The single-agent retrieval-reasoning-verification loop is broken down into five distinct agents with specific functions, so as to allow each agent to be optimized for its specific subtask of the loop without the competing goals that discourage individual agent performance.

Notably, the confidence score gating mechanism employed by the Response Agent is important for considerations of practical deployment. The system guarantees reliable results and ensures that low-confidence answers are not output until they can be retrieved again or through human review, which is not guaranteed in normal RAG/ LLM use cases. The empirical correlation between the aggregated confidence score C and observed hallucination rate (Pearson r across the evaluation set of 0.91) is used to validate the reliability of confidence score as a proxy for response reliability that can be exposed to end users for informed trust calibration.

The total latency of 300 ms for RAMA compared with the Standard LLM baseline is mostly due to the entailment classification of the Verification Agent (about 180 ms) and the knowledge graph look-up when verifying entities (about 95 ms). Batch processing of multiple claim verifications, caching of search results of the knowledge graph query result for frequently queried entity pairs can be optimized. Early experimentation with these optimizations indicates the RAMA can match the Standard LLM baseline with RAMA latency of about 1400 ms, while still maintaining high verification quality.

8. Conclusion and Future Scope

The authors introduced a new approach called RAMA (Retrieval-Augmented Multi-Agent Architecture) for reducing hallucinations in large language models (LLMs) in this paper. To ensure LLM outputs are grounded in validated knowledge from dense-vector corpora and structured knowledge graphs, RAMA uses five specialized agents- Planner, Retrieval, Reasoning, Verification, and Response- that operate via set conventions of communication. When combined with a threshold-based gating mechanism implemented by the Response Agent, the Verification Agent's claim-level entailment checking coupled with a threshold-based gating mechanism by the Response Agent make it possible to obtain a reliable guarantee that is not otherwise offered with currently known approaches to RAG.

Experimental evaluations were performed on two kinds of benchmarks: HotpotQA, TriviaQA, and Natural Questions, with the hallucination rate of 4%, question-answering accuracy of 97%, and AUC of 0.97, resulting in state-of-the-art performance on all evaluated metrics. Overall, these findings attest to the fact that multi-agent collaborative verification can be shown to provide systematic gains in improvement over standard RAG and single-agent RAG architectures, while still having a latency penalty that is tolerable in most “real-world” high-stakes deployment scenarios where factual reliability matters.

References

[1]. P. Lewis, E. Perez, A. Piktus et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. NeurIPS, vol. 33, pp. 9459–9474, 2020.

[2]. S. Mishra, G. Mishra, H. Manob, M. Maurya, K. S. Abhinit, and P. K. D, “Optimized speed control strategies for BLDC motors in Solar-Powered grass Cutting applications through Field-Oriented Control,” International Journal of Research and Development in Engineering Sciences, vol. 6, no. 4, p. 20, Jul. 2024, doi: 10.63328/ijrdes-v6ri4p4.

[3]. N Rama Kumar , A Heena Kousar , M Jahnavi , P Lokeswari , K Harshitha , “Compression of Depper Images for Hybrid Contexts of Picture Order and Recreation ” ,International Journal of Computational Science and Engineering Research, vol. 1, no. 3, p. 28, July. 2024, doi: https://doi.org/10.63328/IJCSER-V1RI3P7

[4]. D Ramachandra Reddy , M Mohammed Fazil Khan , Farooq , T Mahesh , M Nagendra Babu, “Prediction of Machine Failure Status using Machine Learning Techniques” ,International Journal of Computational Science and Engineering Research, vol. 1, no. 3, p. 9, July. 2024, doi: https://doi.org/10.63328/IJCSER-V1RI3P3

[5]. R Nivedha , P Saalim , T Naveen , S Noorshavalli , Y Uday Kiran, “Customer Profiling Method for Hi-Tech Trading Apps” ,International Journal of Computational Science and Engineering Research, vol. 1, no. 3, p. 6, July. 2024, doi: https://doi.org/10.63328/IJCSER-V1RI3P2

[6]. Chaitanya Naick , Reddy Prasad , Affarn Khan , Murali Krishna, “Assessing Fluoride And Chloride Levels In Punganur Water Resources: A Comprehensive Experimental Investigation ”

,International Journal of Computational Science and Engineering Research, vol. 1, no. 3, p. 1, July. 2024, doi: https://doi.org/10.63328/IJCSER-V1RI3P1

[7]. 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

[8]. K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "REALM: Retrieval-augmented language model pre-training," in Proc. ICML, pp. 3929–3938, 2020.

[9]. D. K. D and M. G, "Neurodegenerative Disorder Detection using CNN," International Journal of Research and Development in Engineering Sciences, vol. 6, no. 2, p. 20, Apr. 2024, doi: 10.63328/ijrdes-v6ri2p5.

[10]. S Muni Ratnam , P Bhavitha , P Asma Khanum , S Madhavi , M Bharathi, "Identification of Paddy Leaves Disease using Convolutional Neural Networks," International Journal of Computational Science and Engineering Research, vol. 1, no. 4, p. 66, Nov. 2025, doi: 10.63328/ijcser-v1ri4p13

[11]. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in Proc. UIST, pp. 1–22, 2023.

[12]. V. Karpukhin, B. Oguz, S. Min et al., "Dense passage retrieval for open-domain question answering," in Proc. EMNLP, pp. 6769–6781, 2020.

[13]. S Jaffer shareef , C Sai Prathyusha , S. Shaheen , P Azhar Ali Khan , S Bhuvaneswari , M Reddy Vamsi , B Surendhra , "Analysis and Implementation of a Switching Bi-Directional BUCK-BOOST Conerter for the HEV Applications by using FLC" , International Journal of Computational Science and Engineering Research, vol. 1, no. 4, p. 80, Nov. 2025, doi: 10.63328/ijcser-v1ri4p15

[14]. C. S. Pakala, R. V. M, V. C, P. K. P, and P. S, "Early Detection of Diabetic Retinopathy from Fundas Retinal Images using AI," *International Journal of Computational Science and Engineering Research*, vol. 1, no. 4, p. 40, Oct. 2025, doi: 10.63328/ijcser-v1ri4p8.

[15]. J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Trans. Big Data, vol. 7, no. 3, pp. 535–547, 2021.

[16]. X. Ye, N. Durrett, "The unreliability of explanations in few-shot prompting for textual reasoning," in Proc. NeurIPS, 2022.

[17]. P. K. R, and N. P, "Hydrological Monitoring using Internet of Things," *International Journal of Computational Science and Engineering Research*, vol. 1, no. 4, p. 10, Oct. 2025, doi: 10.63328/ijcser-v1ri4p3.

[18]. 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

[19]. J. Wei, X. Wang, D. Schuurmans et al., "Chain-of-thought prompting elicits reasoning in large language models," in Proc. NeurIPS, vol. 35, pp. 24824–24837, 2022.

[20]. X. Wang, J. Wei, D. Schuurmans et al., "Self-consistency improves chain of thought reasoning in language models," in Proc. ICLR, 2023.

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:

Ravi Samikannu, Retrieval-Augmented Multi-Agent Architecture for Hallucination Reduction in Large Language Models, International Journal of Computer Science , Engineering and Artificial Intelligence , Vol. 2, Issue. 4 (October – December) , 2025 , Pages: 7 – 13.
Crossref DOI : https://doi.org/10.63328/IJCSEAI-V2RI4P2