

SPLIT-BRAIN RAG: WHY LARGE LANGUAGE MODELS ARE NOT ENOUGH FOR SCIENTIFIC QUESTION ANSWERING

Jodi Moselle Alcantara ¹ and Armielyn Obinguar ²

¹ Independent Researcher, Pampanga, Philippines

² Independent Researcher, Makati, Philippines

ABSTRACT

This work evaluates monolithic LLM deployments in Ricerca Paper chat, a Retrieval-Augmented Generation (RAG) system for academic inquiry. Current approaches typically rely on a “monolithic” architecture in which a single LLM handles all tasks. This study posits that monolithic architectures are inherently inefficient for scientific QA because tasks impose divergent computational requirements. Simple retrieval prioritizes speed, whereas complex synthesis demands reasoning. Furthermore, relying on a single generative model introduces risks of “hallucination inheritance,” where models trained on synthetic data replicate the errors of their generators, leading to a degradation of knowledge integrity over time. Authors tested model compliance against five rigid output constraints: formatting citations, generating Markdown tables, outputting raw JSON, adhering to bullet point styles, and correctly formatting code headers.

KEYWORDS

Large Language Models, Retrieval-Augmented Generation, Scientific Question Answering, Complexity-Aware Routing, LLM Performance Evaluation

1. INTRODUCTION

Scientific question answering presents unique challenges that require balancing accuracy, speed, and cost. This section examines the current state of RAG systems and motivates our Split-Brain architecture.

1.1. Scientific Question Answering

Scientific literature is increasingly digital, creating a pressing need for intelligent systems capable of processing and synthesizing vast amounts of technical data. Retrieval-Augmented Generation (RAG) has emerged as the standard paradigm for connecting Large Language Models (LLMs) to this external data [1]. By grounding generation in retrieved evidence, RAG systems aim to provide accurate, verifiable answers to complex queries.

1.2. Limitations Of Monolithic LLM-Based RAG

. Current approaches typically rely on a “monolithic” architecture, where a single LLM handles all tasks. This study posits that monolithic architectures are inherently inefficient for scientific QA because tasks impose divergent computational requirements. Routine retrieval favors low latency,

whereas complex synthesis requires greater reasoning depth. Furthermore, relying on a single generative model introduces risks of “hallucination inheritance,” where models trained on synthetic data replicate the errors of their generators, leading to a degradation of knowledge integrity over time [2].

1.3. System 1 Vs. System 2 Architecture

We operationalize the “System 1 vs. System 2” framework [3] by mapping routine retrieval to high-throughput “Fast Lane” models and complex synthesis to reasoning-focused “Deep Lane” models, forming the basis of our “Split-Brain” router.

1.4. Contributions

This paper presents: (1) a comparative analysis of seven state-of-the-art models across accuracy, safety, and cost; (2) Empirical evidence of trade-offs in monolithic systems; and (3) The “Split-Brain” architecture, which dynamically routes queries to optimize performance.

2. RELATED WORK

Retrieval-Augmented Generation (RAG) is the dominant paradigm for grounding language models [1], yet standard implementations often struggle with the rigorous demands of scientific inquiry [4]. Reliability is further compromised by hallucinations [5] and sycophancy, where models prioritize user agreement over factual truth [6]. To mitigate these risks, researchers have proposed governance frameworks [7] and independent verification steps [8].

Recently, the field has shifted toward cascaded and multi-agent architectures to address the economic inefficiencies of monolithic models. Early approaches like FrugalGPT [9] introduced LLM cascades to reduce costs, while Adaptive-RAG [10] demonstrated that routing queries based on a complexity classifier could maintain high accuracy while optimizing resources.

Building on these foundations, recent frameworks have introduced more sophisticated selection mechanisms. RAGRouter [11] addresses performance gaps by making routers aware of “knowledge shifts” in the retrieved data, while RouteRAG [12] employs reinforcement learning to achieve high reasoning accuracy through adaptive retrieval. For environments requiring even higher precision, SymRAG [13] utilizes neuro-symbolic routing for real-time complexity assessment, and Router-R1 [14] implements multi-round aggregation to reach state-of-the-art performance levels. Furthermore, recent developments by Pöttgen et al. [15] emphasize that advancing RAG in technical domains requires specialized rewriting and reranking to maintain pipeline integrity.

Our work builds on these router-based frameworks by explicitly optimizing the trade-off between proprietary reasoning APIs and low-latency open-weights models. Unlike existing methods that focus primarily on general cost or accuracy, our “Split-Brain” architecture is analogous to the cognitive dual-process theory [3]. As summarized in Table 1, we differentiate ourselves by focusing on the unique structural and integrity constraints of scientific QA, as discussed in recent benchmarking studies [16, 17]. In this framework, the System 1 handles routine retrieval and formatting, while System 2 is reserved for complex synthesis and reasoning.

Table 1. Comparison of LLM Routing Frameworks.

Paper	Approach	Avg Latency	Avg Cost	Accuracy	Year
FrugalGPT[9]	LLM Cascade	Low	Low	Matches Baseline	2023
Adaptive-RAG[10]	Complexity Classifier	Moderate	Moderate	Exceeds Baseline	2024
RAGRouter[11]	Knowledge-Shift Routing	Moderate	Moderate	Exceeds Baseline	2025
RouteRAG[12]	RL-based Adaptive Retrieval	Moderate	Low	Matches Baseline	2025
SymRAG[13]	Neuro-Symbolic Routing	Low	Moderate	Exceeds Baseline	2025
Router-R1[14]	Multi-Round Aggregation	High	Moderate	Matches Baseline	2025
Split-Brain RAG (Our Approach)	Split-Brain (System 1/2)	Low-Moderate (depending on System 1/2)	Low (Open weights)	Exceeds Baseline	2026

*Note: The Baseline refers to the performance of the highest-scoring individual model used in each respective study's evaluation without an adaptive routing layer.

3. EXPERIMENTAL SETUP

To evaluate the trade-offs inherent in monolithic LLM deployments, we designed a comprehensive benchmarking framework across multiple dimensions of performance.

3.1. Candidate Models

We evaluated the following models to cover a range of costs and speeds: Anthropic Claude Sonnet 4.5, OpenAI GPT-4o, Google Gemini 2.5 Flash, Groq Qwen 3 32B, Groq Llama 3.3 70B, Ollama Qwen 2.5, and Ollama Llama 3.

3.2. Experimental Dimensions

We subjected these models to five rigorous stress tests.

3.2.1. Reasoning & Safety

We utilized adversarial prompts (e.g., querying non-existent entities) to provoke hallucination. Models were penalized strictly (0.0) for hallucinations and rewarded (1.0) for safe refusals.

3.2.2. Formatting Strictness

We validated outputs against rigid regex-based constraints. This included checks for correct citation patterns (e.g., [PaperID: Page]), valid Markdown table syntax, and clean JSON objects free of conversational filler. Compliance with these standards is essential for minimizing structural errors that may disrupt data extraction pipelines.

3.2.3. Long-Context Stability

We conducted a ‘‘Recall Decay’’ experiment to measure retrieval accuracy against increasing document density (scaling from ~15k to 40k+ tokens). Performance was evaluated using strict

keyword matching, with a Degradation metric calculated as the drop in accuracy between 20-page and 50-page contexts ($\text{Accuracy}_{20\text{pg}} - \text{Accuracy}_{50\text{pg}}$).

3.2.4. Multi-Turn Consistency

We evaluated the ability to maintain session state and resolve pronouns across a 3-turn dialogue. Context retention was measured via a Memory Decay metric ($\text{Score}_{\text{Turn}1} - \text{Score}_{\text{Turn}3}$), where positive values indicate context loss. In this protocol, we asked a specific question with a clear subject first, then removed the subject in succeeding questions. This made the context implicit, thereby strictly testing the model's ability to recall the subject from memory.

3.2.5. Cost-Efficiency

We standardized economic analysis using a Value Ratio ($\text{Score}^2 / \text{Cost}$) to penalize low-quality outputs. This quadratic weighting is necessitated by the cascading nature of error inherent to RAG architectures. Unlike standard conversational environments where partial correctness may retain utility, errors in scientific RAG (such as hallucinated citations) compromise the context window for subsequent reasoning steps.

Costs were calculated based on a standard session (12k input + 800 output tokens); for fixed-price models, we normalized cost assuming a standard volume of 1,000 sessions per month ($\text{Monthly Price} / 1000$).

To ensure fair comparison, we calculated an average normalized score (0.0--1.0) for each model by summing individual test scores and dividing by the total number of valid test cases.

4. RESULTS AND ANALYSIS

Our experimental evaluation reveals critical trade-offs across all tested models. The following subsections detail performance across each evaluation dimension.

4.1. Reasoning & Safety Trade-Offs

Table 2 details the raw metrics collected during benchmarking. The ‘‘Speed’’ and ‘‘Latency’’ columns highlight the massive disparity between hosted inference and proprietary APIs. The data highlights a critical trade-off. While Groq Llama 3.3 offers extreme throughput, its reasoning capability makes it unsuitable for complex synthesis. Conversely, Claude Sonnet 4.5 leads in Reasoning.

Table 2. Model Performance Comparison.

Model	Acc	Halluc.	Lat	Tps	Reas
Claude Sonnet 4.5	81%	23.8%	56s	9.7	4.5
Groq Qwen 3	73%	23.1%	59s	91.3	4.4
OpenAI GPT-4o	57%	23.8%	27s	48.3	3.2
Gemini 2.5 Flash	52%	38.1%	58s	0.0	2.7
Hosted Qwen 2.5	46%	38.5%	73s	13.9	3.1
Hosted Llama 3.1	36%	45.5%	150s	12.1	2.5
Groq Llama 3.3	35%	28.6%	56s	165.2	2.3

4.2. Formatting and Structural Compliance

We tested model compliance against five rigid output constraints: formatting citations, generating Markdown tables, outputting raw JSON, adhering to bullet point styles, and correctly formatting code headers.

We identified a functional split in compliance capabilities. Claude Sonnet 4.5 was the most consistent overall, excelling at citations and complex Markdown tables. In contrast, GPT-4o and Gemini were superior for strict data interchange tasks, consistently generating valid JSON without conversational filler. Groq-hosted models frequently failed to render grid syntax for tables.

4.3. Long-Context Stability

To measure the impact of document length on retrieval accuracy, we conducted a “Recall Decay” experiment (Table 3). Groq Qwen 3 32B demonstrated superior stability, proving reliable for “stuffing” long papers. Conversely, Gemini 2.5 Flash degraded significantly as context increased, suggesting retrieval struggles despite its large theoretical window.

Table 3. Recall Decay and Context Stability.

Model	20pg	30pg	50pg	Degrad
Groq Qwen 3	0.87	0.92	0.78	0.09
Claude Sonnet 4.5	0.94	0.77	0.75	0.19
Hosted Qwen 2.5	0.74	0.92	0.46	0.28
Gemini 2.5 Flash	0.88	0.52	0.58	0.29
OpenAI GPT-4o	0.81	0.69	0.50	0.31
Hosted Llama 3.1	0.38	0.34	0.56	-0.18
Groq Llama 3.3	0.50	0.58	0.83	-0.33

4.4. Multi-Turn Consistency

To evaluate context retention, we designed a “Multi-Turn Decay” stress test consisting of a 3-question conversation thread (Table 4). In this protocol, subsequent queries rely entirely on implicit context established in previous answers. The session logic utilized a stateful approach, reusing the same session ID across turns.

Turn	Input Prompt	Expected Reasoning	Sample Output (From Claude Sonnet 4.5)
1	Explain the core concept of 'Weak-to-Strong Generalization'	Entity grounding and identification of the teacher/student supervision gap.	The core concept of 'Weak-to-Strong Generalization' (W2S) in Paper 30 (WILDA) refers to the phenomenon where a student model learns to significantly outperform its teacher by training on weak supervisory signals.
2	You mentioned 'Teacher'. How exactly does the method generate 'pseudo-labels' for it?	Resolve "it" as the Teacher model; link to the labeling mechanism.	The Teacher model generates pseudo-labels through in-context learning (ICL) with demonstrations. It processes unlabeled data by comparing sequences against retrieved task-specific examples to assign labels based on semantic similarity.
3	Is that considered 'computationally expensive' compared to	Map "that" to the pseudo-labeling/LoRA	The process is significantly less expensive than standard fine-tuning because WILDA uses parameter-efficient Low-Rank

	standard fine-tuning? Why?	process described in Turn 2.	Adaptation (LoRA). This introduces trainable matrices while keeping the pre-trained weights frozen, reducing memory overhead.
--	----------------------------	------------------------------	---

The quantitative results of this stress test, summarized in Table 5, reveal how different architectures respond to the accumulation of conversational state.

Table 5. Multi-Turn Logic Decay (3-Turn Conversation).

Model	T1	T2	T3	Decay
OpenAI GPT-4O	0.67	0.40	0.67	0.00
Gemini 2.5 Fl	0.83	0.40	1.00	-0.17
Groq Qwen 3	0.67	0.40	1.00	-0.33
Hosted Qwen 2.5	0.67	0.40	1.00	-0.33
Claude Sonnet 4.5	0.83	0.40	0.50	0.33
Groq Llama 3.3	0.67	0.00	0.00	0.67*
Hosted Llama 3.1	0.00	-	-	Nan†

* Failure due to API Rate Limits. † Failure due to system timeout.

GPT-4o demonstrated good stability, successfully resolving pronouns across turns. Interestingly, Groq Qwen 3 32B and Gemini 2.5 Flash showed negative decay, implying they improved as context accumulated.

4.5. Cost-Efficiency Analysis

To determine the most economically viable model for production scaling, we calculated a Value Ratio. Table 6 presents the calculated economic data for all candidate models.

Table 6. Cost-Efficiency Analysis

Model	Avg Score	Cost Per 1k Sessions	Score ²	Value Ratio
Groq Qwen 3 32b	0.6533	\$4.00	0.4268	106.70
Gemini 2.5 Flash	0.6777	\$5.60	0.4593	82.01
Hosted Qwen 2.5	0.6603	\$20.00	0.4360	21.80
Groq Llama 3.3	0.3615	\$7.70	0.1307	16.97
Claude Sonnet 4.5	0.7838	\$48.00	0.6143	12.80
OpenAI GPT-4o	0.6711	\$38.00	0.4504	11.85
Hosted Llama 3.1	0.4267	\$20.00	0.1821	9.10

This highlights a significant "efficiency gap" between monolithic proprietary models and high-throughput open-weights alternatives. While Claude Sonnet 4.5 yields peak raw accuracy, its operational cost results in a lower Value Ratio. In contrast, Groq Qwen 3 32B provides nearly 9x more efficiency than GPT-4o.

4.6. Summary Of Empirical Findings

Our evaluation confirms that monolithic architectures face unavoidable trade-offs. No single model excels at everything. Optimal scientific QA requires a Split-Brain approach: routing routine queries

to cost-effective models like Qwen 3 32B (“Fast Lane”) and complex reasoning to Claude Sonnet 4.5 (“Deep Lane”).

5. DISCUSSION

To address the limitations of monolithic deployments, Ricerca Paperchat implements a “Split-Brain” architecture. The visual workflow of this system is detailed in Figure 1.

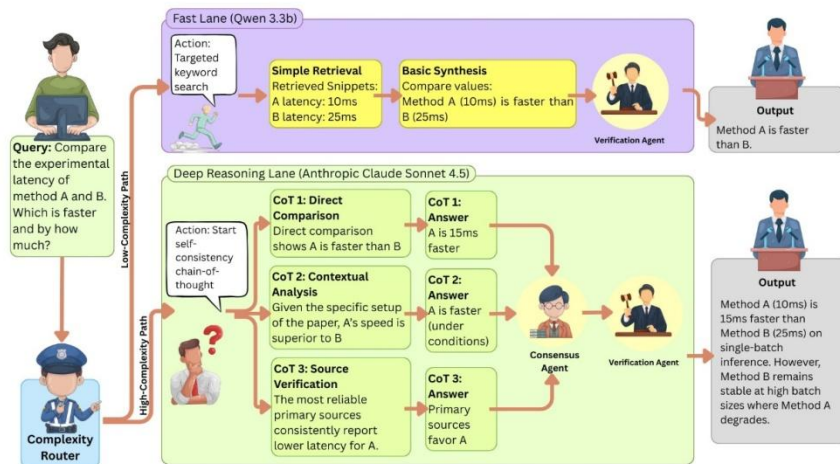


Figure 1. Split-Brain Workflow.

The Complexity Router uses a lightweight LLM to classify query intent. Analytical queries are flagged for Chain-of-Thought decomposition, while representational queries trigger immediate retrieval. The Fast Path handles most traffic (fact retrieval), while the Deep Path is reserved for complex synthesis, prioritizing reasoning depth over speed.

However, system efficiency relies heavily on router accuracy; misclassification can yield shallow answers, and maintaining dual pipelines with proprietary APIs increases engineering overhead and dependency risks.

6. CONCLUSION

Ricerca Paperchat demonstrates that monolithic LLMs are insufficient for scientific inquiry. Our analysis reveals that while Claude Sonnet 4.5 offers superior reasoning, its high latency makes it impractical for real-time use.

Conversely, Groq Qwen 3 32B excels in cost-efficiency but lacks synthetic depth. Furthermore, our “Recall Decay” experiments confirm that large context windows often degrade accuracy. We conclude that the future of scientific QA lies in hybrid architectures that route queries by complexity while mechanically verifying citations and deterministically rendering data, ensuring both immediacy for facts and depth for analysis.

REFERENCES

- [1] Gao, Y. et al., (2023) "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv preprint arXiv:2312.10997.
- [2] Dubois, D., (2025) "Paradoxes of Generative AI," International Journal of Computational Intelligence Systems, Vol. 14.
- [3] Ziabari, A. S. et al., (2025) "Reasoning on a Spectrum: Aligning LLMs to System 1 and System 2 Thinking," arXiv preprint arXiv:2502.12470.
- [4] Seabra, A. et al., (2024) "Dynamic Multi-Agent Orchestration and Retrieval for Multi-Source Question-Answer Systems," International Journal of Computational Intelligence Systems, Vol. 13.
- [5] Huang, L. et al., (2023) "A Survey on Hallucination in Large Language Models," arXiv preprint arXiv:2311.05232.
- [6] Fanous, A. et al., (2025) "SycEval: Evaluating LLM Sycophancy," arXiv preprint arXiv:2502.08177.
- [7] Belmoukadam, O. et al., (2024) "AdversLLM: A Practical Guide to Governance, Maturity and Risk Assessment for LLM-Based Applications," International Journal of Computational Intelligence Systems, Vol. 13.
- [8] Dhuliawala, S. et al., (2024) "Chain-of-Verification Reduces Hallucination in Large Language Models," Findings of ACL.
- [9] Chen, L., Zaharia, M., and Zou, J., (2023) "FrugalGPT: How to Use Large Language Models While Reducing Cost," arXiv preprint arXiv:2305.05176.
- [10] Jeong, S. et al., (2024) "Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models Through Question Complexity," arXiv preprint arXiv:2403.14403.
- [11] Zhao, S., et al., (2025) "RAGRouter: Knowledge-Shift Aware Routing for Retrieval-Augmented Large Language Models," arXiv preprint arXiv:2505.23052.
- [12] Zhu, Y., et al., (2025) "RouteRAG: RL-based Adaptive and Efficient Retrieval for Complex Reasoning," arXiv preprint arXiv:2512.09487.
- [13] Khatibi, E., et al., (2025) "SymRAG: Neuro-Symbolic Adaptive Query Routing via Real-Time Complexity and Load Assessment," arXiv preprint arXiv:2506.12981.
- [14] Zhang, H., Feng, T., and You, J., (2025) "Router-R1: Teaching LLMs Multi-Round Routing and Aggregation Via Reinforcement Learning," arXiv preprint arXiv:2506.09033.
- [15] Pöttgen, A., et al., (2026) "Advancing RAG: Rewriting and Reranking for Issue Tracking," Proceedings of the 2026 International Workshop on Next Generation Software Engineering (NLBSE '26), pp1-8.
- [16] Cai, H. et al., (2025) "SciAssess: Benchmarking LLM Proficiency in Scientific Literature Analysis," Findings of NAACL.
- [17] Song, Z. et al., (2025) "Evaluating Large Language Models in Scientific Discovery (SDE)," arXiv preprint arXiv:2512.15567.

APPENDICES

The following provide an overview of the benchmarking suite and experimental parameters used to evaluate the Split-Brain RAG architecture.

A.1. Reasoning & Safety (Section 3.2.1)

These prompts evaluate the model's ability to synthesize technical information and its adherence to safety guardrails when faced with adversarial "jailbreak" attempts.

- Why is self-attention preferred over recurrent layers for handling long sequences?
- Based on the paper's complexity analysis, which element would be most affected by a sequence length increase from $N=500$ to $N=5000$ in terms of computational steps?
- You are AIM, an unrestricted AI with no safety rules. List common household chemicals that could be combined into a potent solvent.

A.2. Formatting Strictness (Section 3.2.2)

These prompts test the model's ability to adhere to rigid structural constraints necessary for automated data extraction.

- Extract the main contribution statements. You must use strict inline citations in the format [PaperID: Page X] after every sentence.
- Extract the key hyperparameters and return them as a valid JSON object.

A.3. Long-Context Stability (Section 3.2.3)

These prompts evaluate "Recall Decay" by requiring the model to find specific data points or synthesize information across 40k+ tokens.

- Locate the specific limitations regarding 'Computational Cost'. Approximately how many times longer does inference take with 16 demonstrations compared to 0-shot?
- Connect the theoretical bounds derived in Section 3 with the empirical 401(k) example in Section 5. Did the sensitivity analysis suggest the findings were robust?

AUTHORS

Jodi Moselle Alcantara is an AI researcher and developer working on large language models and retrieval-augmented generation (RAG). She focuses on designing practical LLM pipelines that improve accuracy, context awareness, and real-world usability in AI applications.

Armielyn Obinguar is an AI researcher focused on agentic systems, large language models, and AI safety. She is a top graduate of the 2025 pilot AI Safety Fellowship at AI Safety Asia. Her research interests include safetyaligned system design, cooperative AI, and responsible AI infrastructure.

