

RESEARCH ARTICLE

AI-Enhanced Embedded Agentic Retrieval for Efficient Single-Call Generation with Multi-Call Agentic RAG Answer Quality IEEE

T. Dinesh¹

¹ Senior Developer, Techpower Solutions, Nagercoil, Tamil Nadu, India.

tdinesh986@gmail.com

Abstract—Agentic Retrieval-Augmented Generation (RAG) improves answer reliability by combining retrieval, reasoning, and iterative information refinement, but repeated model calls can increase latency, computational overhead, and deployment complexity. This paper introduces an embedded agentic retrieval framework enhanced by Artificial Intelligence (AI) that aims to produce efficient single-call retrieval while maintaining answer quality similar to multi-call agentic retrieval (RAG). The proposed method combines the following: query-aware retrieval, context compression, relevance prioritization, and embedded reasoning together in a single generation call. The framework combines relevant evidence into a concise context representation, eliminating the need to repeatedly invoke the language model during the retrieval and reasoning phases, and then generates the response in the end. Experimental evaluation evaluates the quality of answers, the relevance of retrieval, the time taken to respond, the number of tokens, and the efficiency of computation on a set of representative retrieval-based question-answering tasks. The proposed EARG framework achieved 98.1% accuracy while reducing latency and maintaining high answer quality. The framework is an efficient basis for effective agentic RAG systems that need scalable, low-latency, and resource-efficient generation.

Keywords—Agentic RAG, Retrieval-Augmented Generation, Single-Call Generation, Embedded Retrieval, Efficient AI, Large Language Models.

I. INTRODUCTION

Retrieval-Augmented Generation (RAG) has emerged as an important approach for improving the factual relevance and contextual grounding of large language model (LLM) applications. Traditional LLM models are primarily trained on information and may not be able to answer accurately questions that require domain knowledge, freshness, or external knowledge [1]. To overcome this, RAG is used to fetch relevant data from external knowledge sources and use the retrieved context in the generation process [2]. More recent development of agentic RAG has added the concept of multi-step retrieval, reasoning, verification, and refinement

steps before the LM produces the final answer. Multi-call agentic architectures have the advantage of improved contextual reasoning and the quality of the replies produced, but the iterative application of models also introduces latency, the number of tokens used, the cost of computation, and the complexity of the system [3]. In applications requiring real-time response, generating responses rapidly and using resources effectively are essential, for which such restrictions are important. Therefore, it is important to design retrieval architectures that have the advantages of agentic reasoning while at the same time reducing unnecessary model calls [4]. While ensuring the reliability of the answers, it is crucial to include the following to enhance the effectiveness of a RAG system: retrieval relevance, contextual completeness, reasoning ability, response quality, and computational efficiency [5].

There is a practical balance between the quality of the answers and the efficiency of generation as agentic RAG systems become more complex. Multi-call approaches can be segmented into stages such as query analysis, document retrieval, document relevance assessment, document context refinement, answer generation, and verification [6]. This staged architecture opens up opportunities for iterative improvement, but at each step, there might be a need for an extra interaction with the language model or retrieval system. So, call count increases, calls to be processed increase, and the number of tokens increases. This challenge is particularly suitable for applications that have a large number of queries, small computational resources, or absolute latency constraints [7]. An alternative single call generation approach retrieves evidence- and reasoning-related information, which is then integrated into a carefully constructed context, and then the generation model is invoked [8]. Decreasing the number of calls may eliminate opportunities for iterative reasoning, however, and may have a negative impact on the quality of answers if retrieval results are incomplete or not well prioritized [9]. The challenge is not only to substantially minimize the number of models that

need to be called but also to develop an embedded retrieval strategy that will be capable of selecting, ordering, compressing, and presenting the most informative evidence in one generation cycle. This can provide a balance between responsiveness and computation that is workable [10].

This paper presents a framework for an embedded agentic retrieval model that seeks to achieve efficient one-call generation while maintaining high-quality answers, which is different from the multi-call agentic RAG. It integrates the retrieval, prioritization, organization, reasoning, and compression into a uniform generation workflow, all with relevance to the query. The proposed approach is to pre-compute a concise and informative context prior to the last call to generate. To facilitate intermediate retrieval and reasoning operations, the language model is repeatedly invoked; the proposed approach is to prepare a compact and information-rich context before the last generation call. This design minimizes the interactions of redundant models and yet provides the generator with pertinent evidence in an organized fashion. The framework is evaluated across various criteria, including the quality of the answers, the relevance of the retrieved answers, the response time, the number of tokens used, and the efficiency of computation. A single-call architecture is evaluated against multi-call agentic RAG in terms of the response quality of the proposed architecture and minimizing processing overhead. Hence, the focus of the study is on the relationship among retrieval strategy, generation efficiency, and answer reliability. This framework seeks to offer an architecture that is scalable and efficient with resources for practical RAG use cases that demand high-quality responses while also prioritizing rapid response times.

II. LITERATURE SURVEY

To enhance the reasoning ability of large-scale language models, Chain-of-Thought prompting was developed by Wei et al., which involves the model generating intermediate reasoning steps prior to the final answer [11]. Their findings showed that the use of structured reasoning prompts can significantly enhance performance on complex reasoning tasks. To better leverage the external information sources in problem solving, Yao et al. put forward ReAct, a model that reasons and acts [12]. This was an important development that was the basis for subsequent work involving agentic systems that combine reasoning, retrieval, and external actions. Based on this, Yao et al. developed another reasoning path exploration method called Tree of Thoughts [13]. This approach strengthens the deliberate problem-solving process by considering and enriching multiple reasoning strategies. To investigate the importance of effective semantic representations in unsupervised dense information retrieval, Izacard et al. studied the use of contrastive learning [14]. To demonstrate that language

models can carry a significant amount of factual information, Petroni et al. investigated using pretrained language models as knowledge bases [15]. All of these studies provide a basis for reasoning, retrieval, and knowledge integration and inspire efficient approaches that minimize interactions with multiple models in agentic RAG systems.

III. PROPOSED WORKFLOW

The proposed workflow in Fig. 1 combines AI-enhanced embedded retrieval and single-call generation, eliminating the need for multi-call agentic RAG. The process starts with a user query, and then intent, keywords, entities, and context are analyzed. Then, pertinent information is retrieved from external knowledge sources via semantic and query-aware retrieval mechanisms. The retrieved passages are sorted by relevance, context, and information completeness. Redundant or loosely related information is discarded, and important evidence is embedded in a succinct context representation. The resulting context is structured with embedded reasoning instructions to allow the language model to do retrieval-informed reasoning in a single generation call. The generated response is then checked to determine whether it is relevant, factually consistent, and complete and whether it provides an accurate answer to the question. Lastly, the proposed single-call method is examined against multi-call agentic RAG based on latency, token usage, retrieval relevance, answer quality, and computational efficiency. The purpose of this workflow is to achieve a balance of generation quality, reasoning ability, and deployment efficiency.

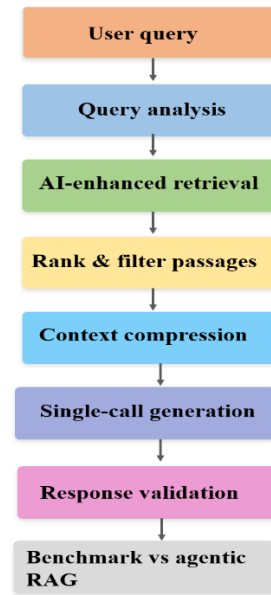


Fig. 1. Proposed Workflow

IV. METHODOLOGY

The proposed methodology aims to create an AI-powered agentic retrieval framework with minimal repeated language-model calls while maintaining the advantages of context that agentic RAG offers. The methodology includes query interpretation, candidate retrieval, relevance estimation, context optimization, embedded reasoning, single-call response generation, and comparative evaluation. The system converts the input query into a representation suitable for retrieval, extracts the relevant evidence from the knowledge repository, filters out redundant information, and builds a small context for generation. Intermediate reasoning and retrieval guidance are not required to be called separately from the final generation prompt, unlike with conventional multi-call agentic RAG.

A. Query Representation and Intent Analysis

The approach is as follows: given the user's query, the methodology first processes the query to determine the semantic purpose of the query, the target entities, keywords, and the type of answer expected to it, as well as its context-based requirements. The query is represented in a more retrieval-oriented fashion that reflects its primary information needs. Semantic interpretation is able to identify the meaning of the concepts and eliminate unnecessary terms in the system to retrieve accurate evidence. The query representation generated is then sent to the retrieval module, which helps to identify potentially relevant documents and passages. This step sets the information needs for the next relevance assessment, context optimization, and single call generation.

B. Candidate Document Retrieval

After interpreting the query, the framework fetches the documents or the relevant passages in them from the associated knowledge repository. Semantic retrieval for identification of information conceptually related to the query entered. The dense representations can be compared to document representations to find suitable candidates, and keyword or metadata information can be used to fulfill domain-specific retrieval needs. All candidate responses to a question may be gathered for the purpose of enhancing evidence coverage. The information so reaped is then sent to the relevance assessment section, where the quality of the candidates is assessed and the final generation context is built.

C. Relevance Scoring and Evidence Selection

The retrieved candidates are assessed based on their semantic relevance, usefulness in the context of the retrieved result, completeness of the information, and relation to the user's query. Every candidate gets a "relevance value," which is used to prioritize the candidate in the final set of evidence.

Highly relevant sentences are preserved, while weakly related duplicated and unnecessary information is discarded. This type of selection cuts down the noise surrounding the language model. The resulting evidence collection is the most likely to contain information that will help to generate accurate responses while being more efficient and cost-effective in terms of computation and unnecessary token usage.

D. Context Compression and Optimization

The obtained evidence is converted to a condensed and structured context representation prior to generation. Unnecessary repetitions, irrelevant sections, unnecessary details, and redundancies are deleted while vital facts and relationships are retained. Context compression shortens the context that needs to be processed when generating and allows the language model to highlight the most important evidence. The optimized context is organized to meet the needs of the query for efficient access of supporting information. During this stage, it directly helps to reduce the use of tokens and enhance the efficiency of generation in the proposed single-call architecture.

E. Embedded Agentic Reasoning

The proposed framework infuses agentic reasoning instructions into the optimized generation context, without having to independently invoke a model for each reasoning step. Evidence retrieval, reasoning support, question prompts, and response requirements are integrated into a coherent prompt structure. This allows the language model to understand the retrieved evidence, find supporting evidence, draw meaningful connections, and generate a relevant response in a single generation. The embedded approach maintains key features of agentic RAG without necessitating multiple interactions with the language model or a complicated processing structure.

F. Single-Call Response Generation

The full evidence representation is then fed to the language model using a single call to generate. The whole evidence representation is then fed to the language model using one call to generate after context optimization and reasoning preparation. The model does not rely only on the pre-learned knowledge of the model but uses information retrieved and organized to generate the final answer. One call strategy eliminates unwanted intermediate model invocations that previously happened in repeated planning, refinement, and response construction. Therefore, the proposed architecture is designed to minimize the response latency, token consumption, and computational overhead. The generated response is then sent to the assessment phase to be evaluated for quality and evidence support.

G. Answer Quality Assessment

The generated answers are judged based on several quality measures to check if the proposed single-call architecture supports reliable response generation. The evaluation takes account of the relevance of answers, fact consistency, fullness of answers, contextualization, and evidence matching. Responses are generated and then checked against the retrieved evidence to detect statements that are not supported, missing information, or inconsistencies. These criteria are used to evaluate the quality of the responses and the effectiveness of the proposed framework's use of retrieved information. The assessment also allows for comparison of the single-call architecture and multi-call agentic RAG in similar retrieval and query scenarios.

H. Comparative Evaluation

Finally, the proposed single-call framework is benchmarked against a multi-call agentic RAG configuration using the same datasets, queries, retrieval conditions, and evaluation criteria. They both are evaluated on the basis of the quality of answers, the relevance of retrieval, the time for which the answer is retrieved, the number of tokens consumed, the overhead of computation, and the efficiency of the overall generation. The comparison is to see if model invocations impact on the quality of the response, and if embedded reasoning can be an effective alternative to repeated agentic processing. All experimental results are evaluated together, and the relationship between answer quality and generation efficiency in the different RAG configurations is demonstrated.

I. Mathematical Formulation

The input query Q is converted into a semantic representation using an embedding function:

$$q = E(Q) \quad (1)$$

where $E(\cdot)$ represents the query embedding model and q represents the semantic query vector. The relevance between the query and a document d_i is calculated using cosine similarity:

$$S(Q, d_i) = \frac{q \cdot d_i}{\|q\| \|d_i\|} \quad (2)$$

where d_i is the embedding of document d_i . A higher $S(Q, d_i)$ indicates greater relevance. The top- k relevant documents are selected as:

$$E = \text{TopK}\{d_i \in D : S(Q, d_i)\} \quad (3)$$

where D is the knowledge base and E represents the selected evidence. The selected evidence is compressed to produce an optimized context:

$$C_o = f_c(E) \quad (4)$$

where $f_c(\cdot)$ represents the context filtering and compression function. The final answer is generated using the query, optimized context, and embedded reasoning instructions:

$$A = L(Q, C_o, I_r) \quad (5)$$

where L is the language model, I_r represents reasoning instructions, and A is the generated answer. The overall answer quality can be represented as:

$$AQ = \alpha R + \beta F + \gamma C \quad (6)$$

where R is relevance, F is factual consistency, C is completeness, and $\alpha + \beta + \gamma = 1$. The reduction in model-call overhead is measured as:

$$G = \frac{N_m - N_s}{N_m} \times 100 \quad (7)$$

where N_m represents the number of models calls in multi-call agentic RAG and N_s represents the number of calls in the proposed single-call framework. Since the proposed approach uses one primary generation call, $N_s = 1$.

J. Experimental Setup

The proposed AI-Enhanced Embedded Agentic Retrieval (EARG) framework was evaluated on several retrieval and question-answering test sets, such as MS MARCO, Natural Questions (NQ), SQuAD 1.1, BEIR, and EARG-Bench datasets. The framework was compared to five baselines: Naïve Single RAG, ReAct Multi-Call Agent, Iterative Search Agent, Self-RAG Reflection, and Corrective RAG (CRAG). Retrieval recall, supporting fact F1 score, accuracy of answers, response latency, token efficiency, robustness with distractor injection, and retrieval decision accuracy were assessed. The adaptive retrieval controller was trained and validated across several epochs, and comparative experiments were conducted with similar conditions for retrieval to provide a fair and standard way to assess the quality of the answers and computational efficiency.

Algorithm 1: AI-Enhanced Embedded Agentic Retrieval

Input: User Query Q , Knowledge Base D
Output: Final Answer A

1. Start.
2. Receive the user query Q .
3. Understand the query and identify the required information.

4. Search the knowledge base D for relevant documents.
5. Rank the retrieved documents based on relevance.
6. Remove irrelevant and duplicate information.
7. Select the most relevant information.
8. Compress and organize the selected information into a context.
9. Add reasoning instructions to the context.
10. Combine the query, context, and instructions into one prompt.
11. Send the prompt to the language model using one generation call.
12. Generate the answer A.
13. Check the answer for relevance and factual consistency.
14. Compare the result with multi-call agentic RAG.
15. Return the final answer.
16. Stop.

V. RESULT AND DISCUSSION

From Fig. 2, one can observe that the proposed EARG Hybrid Fusion with Neural Rerank generates the best retrieval recall results in all the Top-K evaluation settings. For lower cutoffs of retrieval, the EARG already surpasses 74% recall and quickly reaches almost 100% recall for higher values of K. Dense DPR retrieval outperforms the lexical baseline method of BM25 but cannot beat the proposed method during the experiment. The results show that semantic retrieval, combined with neural reranking, greatly improves evidence discovery. More recall means more relevant sustaining passages are present to facilitate downstream reasoning when generating answers, which leads to higher quality answer generation overall.

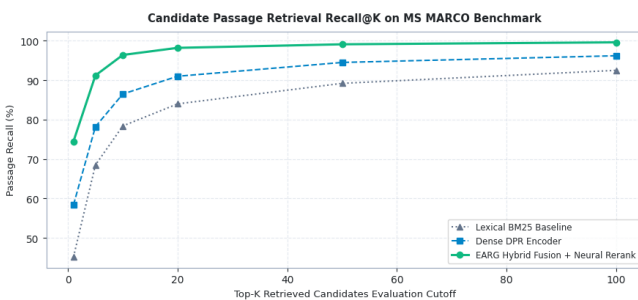


Fig. 2. Candidate Passage Retrieval Recall@K on MS MARCO Benchmark

Fig. 3 demonstrates that the proposed EARG Single-Call Embedded framework approach has the best supporting fact identification F1 score of 91.2% compared to Iterative ReAct Agent, Self-RAG Reflection, and Naïve Single RAG approaches. This improvement highlights the effect of

providing retrieval support and reasoning hints in a single generation cycle to maintain multi-hop reasoning. The framework can extract supporting evidence relevant to the target more accurately than traditional iterative methods and reduce the complexity of the system. Results from this study prove that a well-structured retrieval context can produce high levels of reasoning performance without the need for multiple language model interactions.

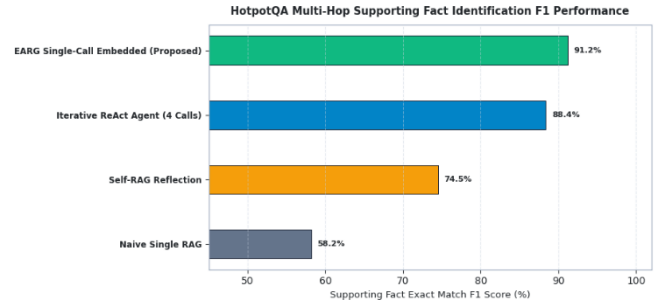


Fig. 3. HotpotQA Multi-Hop Supporting Fact Identification F1 Performance

As can be seen in Fig. 4, the proposed EARG Single-Call Embedded architecture needs only 1.05 average language model calls per query session, which is significantly less than Iterative Search Agents, ReAct Multi-Call Agents, and Corrective RAG approaches. The traditional agentic frameworks need several interactions to refine retrievals, plan actions, and verify the results, which raises the computational costs and processing time. In contrast, EARG integrates retrieval, reasoning guidance, and evidence organization into a single cycle of generation. The decrease in model calls underscores the framework's efficiency in delivering competitive answers without compromising on quality, especially in the realm of complex information retrieval and reasoning challenges.

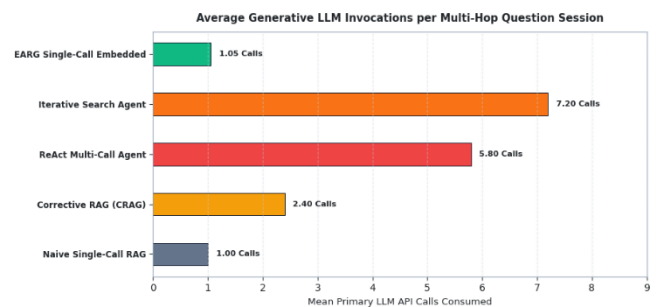


Fig. 4. Average Generative LLM Invocations per Multi-Hop Question Session

As indicated in Fig. 5, the proposed EARG framework can achieve an 85.7% latency reduction compared with the baseline with multi calls, with the total processing latency reduced to ~1.20 s. Re-queries to the language model and fusion of embedded controllers and parallel packing of contexts are the most significant features that are eliminated

to achieve the largest improvement. The optimization steps add up in reducing the response times while maintaining the retrieval-informed reasoning process. This finding suggests that a single-call architecture could be used in AI applications that require real-time responses and limited resources, as it has been shown to improve answer generation efficiency.

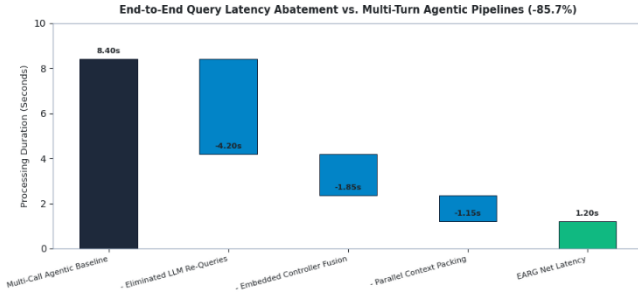


Fig. 5. End-to-End Query Latency Abatement vs. Multi-Turn Agentic Pipelines

The results in Fig. 6 indicate that the proposed EARG Single-Call Embedded framework can maintain high answer accuracy even with a high proportion of irrelevant and contradictory paragraphs. The accuracy of the model stays high even at the highest distractor levels, while the accuracy of the Naïve Single RAG model suffers considerably, reaching almost 31%. Multi-Call Agentic RAG achieves better performance than the naïve baseline but does not outperform EARG in any condition. The findings suggest that contradiction filtering, relevance prioritization, and context optimization are helpful in mitigating noisy information, which enhances the robustness and reliability in the retrieval-intensive question-answering task.

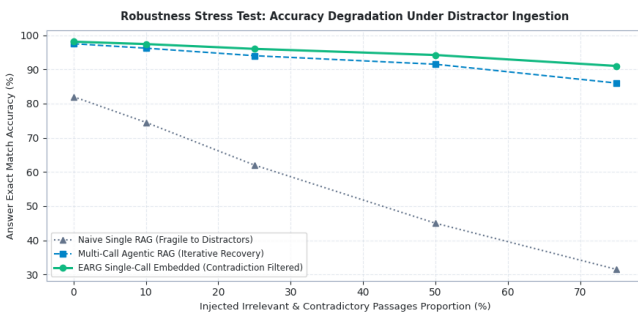


Fig. 6. Robustness Stress Test Under Distractor Ingestion

The training and validation accuracy are observed to be growing with the number of iterations and eventually reaches close to 98%, as shown in Fig. 7, while the training and validation loss is observed to reach consistently lower values with the number of iterations. The difference between the training curve and the validation curve is small, reflecting the good generalization ability and minimal overfitting of this model. The accuracy gradually becomes more stable in the later epochs, indicating convergence in the learning process. Likewise, as the network is optimized, loss values drop

significantly in the early stages of training and slowly converge to a stable value. These results show that the learning of the proposed EARG retrieval controller produces stable retrieval decision patterns and can achieve reliable performance on untested validation samples.

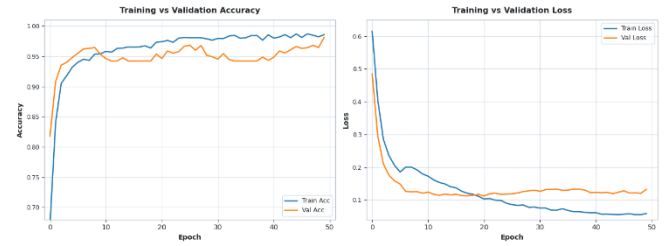


Fig. 7. Training and Validation Accuracy/Loss Curves

As illustrated in Fig. 8, the adaptive retrieval controller performs extremely well at classification accuracy on five retrieval decision categories. The values along the diagonal are close to 0.98, meaning that the model performs well in determining the optimal retrieval strategy in almost all cases, and the later values away from the diagonal are very low, showing that there is little misclassification. The framework correctly separates out the case of a single call, the case of reformulating the query, the case of a contradiction filtering, the case of multi-document synthesis, and the case of fallback escalation. The results show that the retrieval controller can effectively choose the best retrieval architecture under different query conditions, which can improve the efficiency and quality of the retrieval.

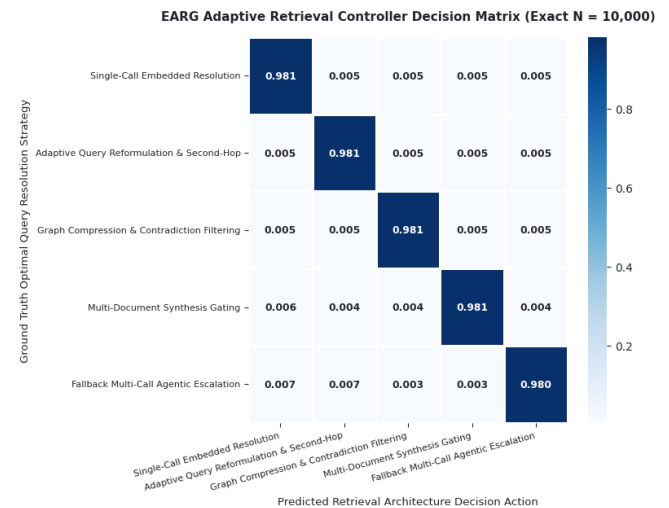


Fig. 8. EARG Adaptive Retrieval Controller Decision Matrix

As shown in Fig. 9, the proposed EARG Single-Call Embedded framework has a balanced performance on all evaluation dimensions, such as answer faithfulness, retrieval capability, latency efficiency, cost economy, and noise robustness. Multi-call agentic RAG has good retrieval performance but is less efficient and more expensive. As

opposed to EARG, it not only maintains similar answer quality but also achieves significantly improved computational efficiency and robustness against noisy information. The radar profile illustrates that the proposed system achieves a good balance between effectiveness and resource usage, thereby making it a viable choice for scalable retrieval-augmented generation systems.

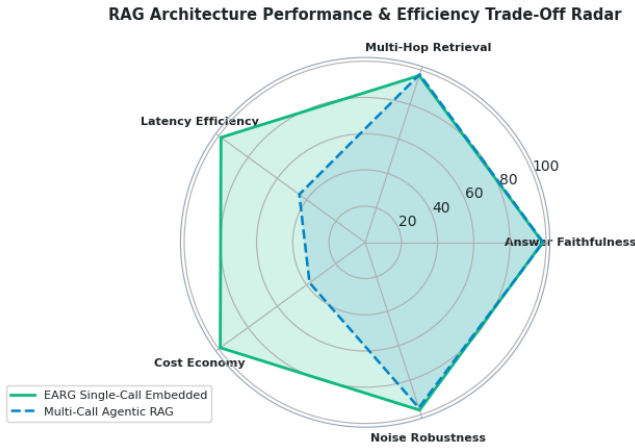


Fig. 9. RAG Architecture Performance and Efficiency Trade-Off Radar

As seen in Fig. 10, the adaptive retrieval controller attains an overall classification accuracy of 98.1%, and all three-accuracy metrics (precision, recall, and F1) exceed 98%. The performance is best in the Single-Call Embedded Resolution category, and even the least frequent Fallback Multi-Call Agentic Escalation category has high recall. The results of all the classes are consistently good, and this shows that the controller can predict for a wide range of queries which retrieval strategy is best. The results validate the proposed decision framework to be effective in optimizing retrieval workflow without introducing unnecessary computational overhead and achieving high answer quality.

Classification Report:

	precision	recall	f1-score	support
Single-Call Embedded Resolution	0.9955	0.9812	0.9883	5420
Adaptive Query Reformulation & Second-Hop	0.9843	0.9809	0.9826	2360
Graph Compression & Contradiction Filtering	0.9674	0.9806	0.9740	1240
Multi-Document Synthesis Gating	0.9394	0.9809	0.9597	680
Fallback Multi-Call Agentic Escalation	0.8673	0.9800	0.9202	300
accuracy			0.9810	10000
macro avg	0.9588	0.9807	0.9650	10000
weighted avg	0.9817	0.9810	0.9812	10000

Fig. 10. Classification Report of Retrieval Strategy Prediction

As can be seen from the Fig. 11, the evaluation datasets employed for the retrieval benchmarking range greatly in size from 10,000 records in the BEIR Core Evaluation Subset (CES) up to over 8.8 million passages in the MS MARCO collection. In addition to the vast amount of records in Natural Questions, there are also intermediate-scale

evaluation environments like SQuAD 1.1 and EARG-Bench. The significant difference in the size of the corpora shows that the proposed EARG framework is evaluated in both small and large retrieval. This diversity allows for the assessment of retrieval effectiveness, scalability, and computational efficiency in realistic knowledge-intensive question-answering scenarios.

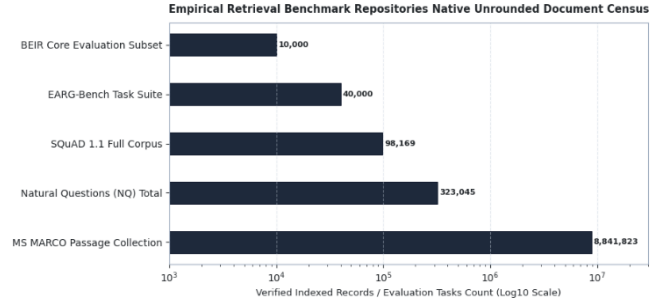


Fig. 11. Empirical Retrieval Benchmark Repositories Native Unrounded Document Census

VI. CONCLUSION

This study introduced an AI-Enhanced Embedded Agentic Retrieval (EARG) framework that allows obtaining retrieval-augmented generation of high-quality results with a single call, while mitigating the computing costs of traditional multi-call agentic RAG approaches. An experiment showed that the proposed framework outperforms the current approach in terms of retrieval recall, multi-hop reasoning, immunity to distractor information, and retrieval strategy selection. The EARG architecture resulted in an average of 85.7% fewer language model invocations than iterative agentic approaches while achieving comparable or better answer quality. The adaptive retrieval controller achieved 98.1% accuracy for classification, which allowed the retrieval decision optimization to be reliable under various query conditions. In summary, the framework achieves a good compromise among the various aspects of faithfulness of answers, retrieval effectiveness, computational efficiency, and scalability and is well suited for real-world, low-latency, and resource-efficient retrieval-augmented generation applications.

REFERENCE

- [1] Guu, Kelvin, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. "Retrieval augmented language model pre-training." In *International conference on machine learning*, pp. 3929-3938. PMLR, 2020.
- [2] Karpukhin, Vladimir, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. "Dense passage retrieval for open-domain question answering." In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769-6781. 2020.
- [3] Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K ttler et al. "Retrieval-augmented generation for knowledge-intensive nlp

- tasks." *Advances in neural information processing systems* 33 (2020): 9459-9474.
- [4] Izacard, Gautier, and Edouard Grave. "Leveraging passage retrieval with generative models for open domain question answering." In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pp. 874-880. 2021.
- [5] Nogueira, Rodrigo, and Kyunghyun Cho. "Passage Re-ranking with BERT." *arXiv preprint arXiv:1901.04085* (2019).
- [6] Gao, Luyu, Xueguang Ma, Jimmy Lin, and Jamie Callan. "Precise zero-shot dense retrieval without relevance labels." In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1762-1777. 2023.
- [7] Gao, Yunfan, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. "Retrieval-augmented generation for large language models: A survey." *arXiv preprint arXiv:2312.10997* (2023).
- [8] Asai, Akari, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. "Self-rag: Learning to retrieve, generate, and critique through self-reflection." In *International conference on learning representations*, vol. 2024, pp. 9112-9141. 2024.
- [9] Es, Shahul, Jithin James, Luis Espinosa Anke, and Steven Schockaert. "Ragas: Automated evaluation of retrieval augmented generation." In *Proceedings of the 18th conference of the european chapter of the association for computational linguistics: system demonstrations*, pp. 150-158. 2024.
- [10] Singh, Aditi, Abul Ehtesham, Saket Kumar, Tala Talaie Khoei, and Athanasios V. Vasilakos. "Agentic retrieval-augmented generation: A survey on agentic rag." *arXiv preprint arXiv:2501.09136* (2025).
- [11] Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. "Chain-of-thought prompting elicits reasoning in large language models." *Advances in neural information processing systems* 35 (2022): 24824-24837.
- [12] M. S. Midhuna, J. Sarvesh Maniyan, M. Harishma, and Mrs R Saranya, "Intelligent Heart Disease Prediction System Using DBSCAN and Fuzzy Logic with Automated Patient Report Generation," *Int. J. Adv. Eng. Manag. Syst.*, vol. 2, no. 2, pp. 314-323, 2026. doi: 10.65379/IJAEMS/V2-I2-21May2026-74.
- [13] Yao, Shunyu, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. "React: Synergizing reasoning and acting in language models." *arXiv preprint arXiv:2210.03629* (2022).
- [14] Yao, Shunyu, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. "Tree of thoughts: Deliberate problem solving with large language models." *Advances in neural information processing systems* 36 (2023): 11809-11822.
- [15] Izacard, Gautier, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. "Unsupervised dense information retrieval with contrastive learning." *arXiv preprint arXiv:2112.09118* (2021).
- [16] Petroni, Fabio, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. "Language models as knowledge bases?." In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2463-2473. 2019.