Task Aware Retrieval Selection Mechanisms for Large Language Model Reasoning

Evelyn T. Chan *, Marcus Y. Lim , Adrian K. Goh *

Posted Date: 2 December 2025

doi: 10.20944/preprints202512.0262.v1

Keywords: task aware retrieval; large language models; bandit optimization; reasoning; adaptive retrieval strategy

 doi:10.20944/preprints202512.0262.v1

Article

Task Aware Retrieval Selection Mechanisms for Large Language Model Reasoning

Evelyn T. Chan *, Marcus Y. Lim and Adrian K. Goh

School of Computing, National University of Singapore, Singapore 119077

* Correspondence: evechan@nus.edu.sg

Abstract

Most retrieval augmented generation pipelines rely on a fixed retrieval configuration across tasks, despite the fact that different reasoning problems benefit from distinct retrieval depth, granularity, and timing. This work introduces task aware retrieval selection mechanisms that adapt retrieval behavior to the characteristics of each query and downstream objective. We design a two level controller: a query level policy network that predicts retrieval depth, document granularity, and whether to perform single shot or iterative retrieval, and a step level policy that can trigger additional retrieval based on intermediate model uncertainty. Both policies are parameterized by lightweight transformers that consume query embeddings, preliminary generation traces, and calibration features such as entropy and self consistency variance. We train the policies with an off policy contextual bandit objective using logged interactions from 1.2 million queries covering factoid QA, multi hop reasoning, and code explanation tasks. When plugged into a standard RAG pipeline with a 13B language model, the task aware mechanism improves overall Exact Match by 3.7 points and reasoning success rate by 5.4 points on StrategyQA, HotpotQA, and GSM8K style synthetic math QA, while reducing average retrieval calls by 18.2%. Detailed analysis shows that the policy learns to avoid unnecessary retrieval for simple questions and to allocate more retrieval budget to compositional and numerically intensive problems.

Keywords: task aware retrieval; large language models; bandit optimization; reasoning; adaptive retrieval strategy

1. Introduction

Large language models (LLMs) are widely used in question answering, coding assistance, and mathematical reasoning, yet many tasks still rely on information that is not encoded in their internal parameters. Retrieval-augmented generation (RAG) addresses this limitation by enabling models to access external documents during inference [1]. Recent work in scientific, medical, and educational applications shows that RAG improves factual accuracy and reduces unsupported statements, but it also reveals that different tasks benefit from different retrieval configurations, including the number of documents retrieved and the timing of retrieval within the reasoning process [2]. Most existing RAG systems nonetheless apply a single fixed retrieval configuration to all inputs. They typically rely on one top-k value and a fixed passage length, following a "retrieve-once-then-generate'' workflow. Although this approach works for simple queries, it becomes unreliable for tasks that require multi-step reasoning or dynamically evolving information needs. Several benchmark analyses show that fixed retrieval rules frequently miss essential evidence or introduce excessive text that distracts the model and disrupts the reasoning chain [3]. To mitigate these failures, recent studies explore adaptive or task-aware retrieval mechanisms. Some methods leverage task descriptions to guide the selection of relevant documents and demonstrate that retrieval behavior varies with user intent [4]. Other approaches introduce self-monitoring, where the model inspects its intermediate outputs and requests additional retrieval when signs of uncertainty appear [5]. A complementary development is the emergence of context-reconstruction modules such as the plug-in reconstructor proposed in [6],

 doi:10.20944/preprints202512.0262.v1

which shows that reorganizing and rewriting retrieved evidence before generation can substantially reduce hallucination. Additional adaptive retrieval strategies rely on confidence-based triggers that skip retrieval for simple inputs while activating it for more complex ones [7]. Bandit-based approaches treat retrieval decisions as actions and learn policies that maximize accuracy while controlling retrieval cost [8]. Related work on document routing and task-aware optimization in LLM–database pipelines similarly reports that adaptive retrieval improves both answer quality and computational efficiency [9,10]. Despite these advances, several open problems remain. Many adaptive retrieval systems modify only a single retrieval factor—typically top-k—without coordinating other important decisions, such as retrieval depth, passage size, or retrieval timing [11]. Most approaches also depend solely on static query features and fail to incorporate signals that arise during generation, such as entropy changes, disagreement across self-consistent samples, or shifts in partial reasoning steps. Generalization is another challenge: models trained on narrow datasets often struggle in mixed workloads that combine fact-based queries, multi-step reasoning, and code explanation tasks [12]. Reasoning benchmarks further illustrate why dynamic retrieval is necessary. StrategyQA consists of short queries requiring implicit multi-hop reasoning, making it difficult to determine retrieval needs in advance [13]. Multi-step math benchmarks such as GSM8K show that irrelevant retrieved documents can break the reasoning chain, indicating that selective retrieval at intermediate steps is essential for maintaining coherent reasoning trajectories [14,15]. A growing body of research highlights the importance not only of making retrieval decisions adaptively, but also of managing the structure and quality of retrieved context. Studies of context organization demonstrate that poorly structured or excessively long contexts can distort reasoning, even when retrieval is accurate. Context reconstruction, evidence rewriting, and span-level filtering have shown promise in stabilizing model behavior and reducing hallucinations by ensuring that retrieved information is presented in a more consistent and interpretable form. These findings imply that retrieval and context organization should be treated as integrated processes, particularly for tasks involving multi-step reasoning.

This study examines task-aware retrieval selection for large-scale reasoning. We introduce a two-level controller that adapts retrieval behavior at both the query level and the step level. At the query level, a policy selects retrieval depth, passage size, and whether retrieval occurs once or across multiple rounds. At the step level, a second policy triggers additional retrieval when intermediate signals—such as entropy, variance across self-consistent samples, or early reasoning traces—indicate elevated uncertainty. Both controllers employ lightweight transformers and take as input the query embedding, partial-generation outputs, and simple calibration signals. The system is trained using an off-policy contextual bandit objective with 1.2 million logged interactions covering factoid QA, multi-hop reasoning, and code-explanation tasks. When integrated into a 13-billion-parameter RAG pipeline, the proposed framework improves exact-match accuracy, increases reasoning success rates, and reduces unnecessary retrieval calls. These results demonstrate that retrieval decisions can be shaped jointly by query characteristics and intermediate reasoning signals, offering a practical path toward more efficient, more adaptive, and more reliable RAG systems.

2. Materials and Methods

2.1. Study Dataset and Task Composition

This study uses a dataset of 1.2 million queries collected from three major task categories: factual question answering, multi-step reasoning, and code-related explanation tasks. The queries come from public benchmarks and anonymized service logs. Each query includes the original text, the retrieval results, and the model's prior outputs. The dataset shows large variation in length and structure, ranging from short factual prompts to long reasoning chains. All samples were screened to remove incomplete entries and duplicated traces. After cleaning, the final dataset reflects a broad range of reasoning patterns needed to train task-aware retrieval selection policies.

 doi:10.20944/preprints202512.0262.v1

2.2. Experimental Setup and Baseline Systems

To examine the effect of adaptive retrieval, we compare the proposed method with two baseline systems. The first baseline uses fixed retrieval settings: a constant retrieval depth, uniform passage size, and a single retrieval stage before generation. The second baseline uses iterative retrieval but without any task-aware adjustment. The proposed system includes a two-level controller that can adjust retrieval depth, choose passage size, and decide when to request additional retrieval. All models share the same retriever and the same 13B language model. This design ensures that the comparison focuses on retrieval behavior rather than differences in model capacity. Each system is evaluated under identical conditions, including hardware, memory limits, and batch scheduling.

2.3. Evaluation Procedures and Quality Checks

For each query, the system records the retrieved documents, intermediate reasoning steps, and the final answer. Accuracy is measured using Exact Match and task-specific success rates. All experiments are repeated three times with different seeds to reduce the effect of random variation. Queries that produce inconsistent logs or missing retrieval records are examined and rerun when needed. During policy training, gradient updates are checked for abnormal values, and early stopping is applied to avoid overfitting. Retrieval outputs are stored separately in a controlled format to maintain consistent evaluation. These steps ensure that all model comparisons are based on reproducible and reliable data.

2.4. Data Processing and Model Equations

Before policy training, each query is converted into embeddings using the base language model. Intermediate signals such as entropy, token variance, and step counts are extracted from generation traces. These features serve as inputs to the policy networks. We use two simple equations to summarize model behavior.

The first equation is a regression model describing the relationship between retrieval depth and answer accuracy [16]:

$$A=\alpha_0+\alpha_1 d+\epsilon,$$

where A is the accuracy, d is retrieval depth, and ϵ is the error term.

The second equation defines the retrieval cost per correct answer:

$$C=\frac{R}{A_c},$$

where R is the number of retrieval calls and A_c is the number of correct answers. This measure helps compare systems that reach similar accuracy but differ in retrieval use. All processing steps follow fixed preprocessing rules to maintain consistency.

2.5. Policy Training and Optimization Strategy

The retrieval-selection policies are trained using an off-policy contextual bandit approach. Logged interactions provide the reward signals without requiring new retrieval calls during training. The policies are implemented as lightweight transformer modules with reduced hidden dimensions. Training batches mix queries from all task categories to prevent bias toward one type of reasoning. A replay buffer is maintained to retain rare long-form queries, which are important for learning when deeper retrieval is needed. Training continues until validation performance stabilizes. After policy convergence, all systems are tested on a held-out set that mirrors the distribution of the training queries.

3. Results and Discussion

3.1. Overall Performance and Retrieval Cost

Across all datasets, the task-aware retrieval method improves both accuracy and retrieval efficiency. When combined with the 13B model, Exact Match rises by 3.7 points, and reasoning

 doi:10.20944/preprints202512.0262.v1

success increases by 5.4 points on StrategyQA, HotpotQA, and GSM8K-style math tasks. At the same time, the average number of retrieval calls falls by 18.2%. These results show that changing retrieval depth and timing for each query is more useful than using one fixed rule.

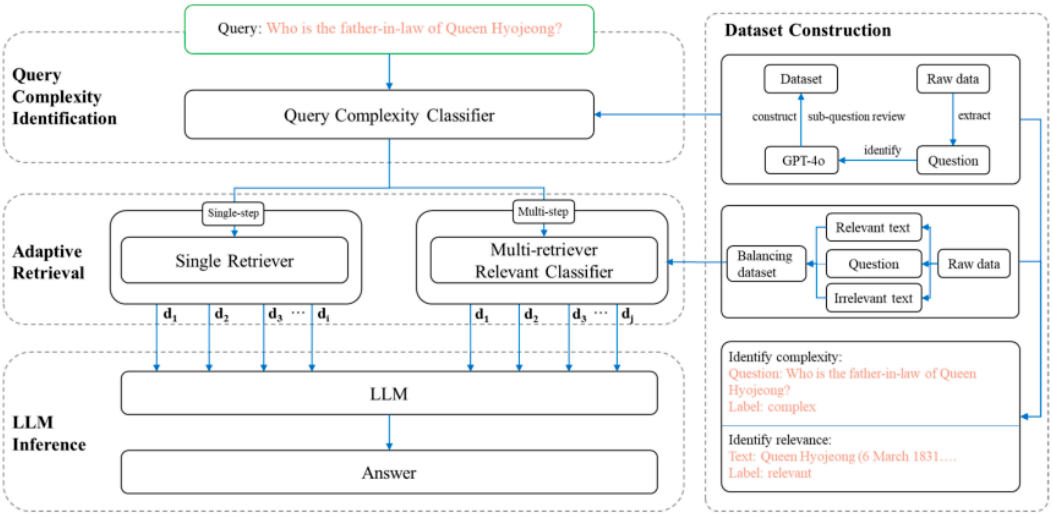

Figure 1. Diagram of the task-aware retrieval process and its main steps.

3.2. Differences Across Task Types

The improvements vary across task categories. For StrategyQA, the controller often selects shallow retrieval for simple questions and deeper retrieval for questions that contain hidden reasoning steps. For HotpotQA, which needs information from multiple places, the controller tends to use longer passages and allows retrieval at more than one point in the reasoning process. This helps the model keep important facts that link different parts of the question. For GSM8K-style math questions, the controller usually turns retrieval off for short arithmetic problems but uses retrieval when the question includes unfamiliar terms or extra factual information. Similar task-dependent patterns have been reported in other RAG systems tested on mixed workloads in QA and scientific domains [17,18].

3.3. Retrieval Use and Policy Decisions

We also examine how the two-level policy uses retrieval. The query-level policy sets an initial retrieval depth and passage length based on the expected difficulty of the query. The step-level policy then adds or skips retrieval according to uncertainty measures such as entropy and variation across sample outputs. For questions in the easiest group, more than 60% require no retrieval beyond the first call. For harder questions, this share drops below 25%, and the controller often adds one more retrieval step during the middle of the reasoning process. This pattern agrees with observations from earlier RAG applications in health and safety studies, where retrieval is used only when internal signals show that more evidence is needed [19].

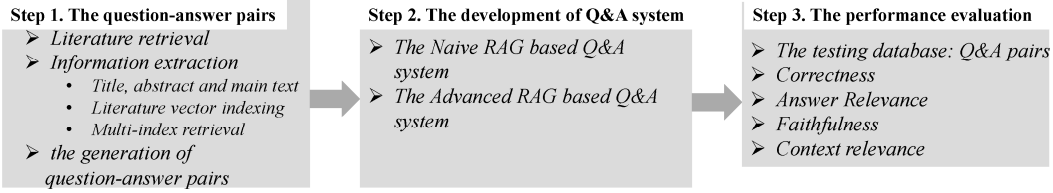

Figure 2. Relation between retrieval use and answer accuracy across different settings.

 doi:10.20944/preprints202512.0262.v1

3.4. Comparison with Related Methods and Limitations

Compared with recent adaptive retrieval approaches, the proposed method offers a simpler and more flexible structure. Bandit-based systems choose from a small set of fixed retrieval options and do not adjust retrieval during reasoning. Other methods depend on multi-stage retrieval graphs that are harder to extend when new tasks appear [20]. In contrast, our method uses compact transformer policies and a single set of shared features, allowing both depth and timing to change within the same model. The results show clear gains with only a small number of extra parameters. However, several limits remain. The method still depends on the quality of the top-ranked retrieved documents, and performance may drop if the retriever returns mostly irrelevant content. In addition, the policies are trained on a fixed set of tasks, and applying them to new domains may require additional data or adaptation. Future work may focus on improving retriever quality, allowing cross-domain transfer, and designing better uncertainty signals for step-level control.

4. Conclusion

This study introduces a retrieval method that adjusts to the needs of each query and each step of reasoning. The results show steady gains in accuracy and a clear reduction in retrieval use when compared with fixed retrieval rules. The two-level design allows the model to use shallow retrieval for simple questions and to request more information only when the reasoning indicates a lack of evidence. These findings suggest that retrieval should be treated as part of the reasoning process rather than a single action taken before generation. The approach can support applications in search assistance, educational tools, scientific writing, and code understanding, where tasks vary in structure and difficulty. However, the method still depends on the quality of retrieved documents, and it may not work well when the retriever provides mostly irrelevant or noisy text. In addition, the policies are trained on a limited set of tasks, so applying them to new domains may require further tuning. Future work may explore better retrieval quality, transfer across domains, and improved signals for detecting uncertainty during reasoning.

References

1. Abo El-Enen, M., Saad, S., & Nazmy, T. (2025). A survey on retrieval-augmentation generation (RAG) models for healthcare applications. Neural Computing and Applications, 37(33), 28191-28267.
2. Maillard, J., Karpukhin, V., Petroni, F., Yih, W. T., Oguz, B., Stoyanov, V., & Ghosh, G. (2021, August). Multi-task retrieval for knowledge-intensive tasks. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 1098-1111).
3. Elbakian, K., & Carton, S. (2025, April). Retrieving Versus Understanding Extractive Evidence in Few-Shot Learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 39, No. 26, pp. 27268-27276).
4. Mussa, O., Rana, O., Goossens, B., Orozco-terWengel, P., & Perera, C. (2024, November). Towards Enhancing Linked Data Retrieval in Conversational UIs Using Large Language Models. In International Conference on Web Information Systems Engineering (pp. 246-261). Singapore: Springer Nature Singapore.
5. Chaudhury, R. (2025). Semi-automated self-monitoring to enhance reflection and awareness among self-directed learners.
6. Li, S., & Ramakrishnan, N. (2025, July). Oreo: A plug-in context reconstructor to enhance retrieval-augmented generation. In Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR) (pp. 238-253).
7. Gumaan, E. (2025). ExpertRAG: Efficient RAG with Mixture of Experts--Optimizing Context Retrieval for Adaptive LLM Responses. arXiv preprint arXiv:2504.08744.
8. Gao, Z., Qu, Y., & Han, Y. (2025). Cross-Lingual Sponsored Search via Dual-Encoder and Graph Neural Networks for Context-Aware Query Translation in Advertising Platforms. arXiv preprint arXiv:2510.22957.

 doi:10.20944/preprints202512.0262.v1

9. Jin, J., Su, Y., & Zhu, X. (2025). SmartMLOps Studio: Design of an LLM-Integrated IDE with Automated MLOps Pipelines for Model Development and Monitoring. arXiv preprint arXiv:2511.01850.
10. Yin, Z., Chen, X., & Zhang, X. (2025). AI-Integrated Decision Support System for Real-Time Market Growth Forecasting and Multi-Source Content Diffusion Analytics. arXiv preprint arXiv:2511.09962.
11. Liang, R., Ye, Z., Liang, Y., & Li, S. (2025). Deep Learning-Based Player Behavior Modeling and Game Interaction System Optimization Research.
12. Guțu, B. M., & Popescu, N. (2024). Exploring data analysis methods in generative models: from Fine-Tuning to RAG implementation. Computers, 13(12), 327.
13. Wu, C., Zhang, F., Chen, H., & Zhu, J. (2025). Design and optimization of low power persistent logging system based on embedded Linux.
14. Ali, Z., & Vadlapati, P. (2025). iRAT: Improved Retrieval-Augmented Thinking for Context-Aware Replanning-Based Reasoning.
15. Zhu, W., Yao, Y., & Yang, J. (2025). Optimizing Financial Risk Control for Multinational Projects: A Joint Framework Based on CVaR-Robust Optimization and Panel Quantile Regression.
16. Wang, J., & Xiao, Y. (2025). Research on Transfer Learning and Algorithm Fairness Calibration in Cross-Market Credit Scoring.
17. Ma, K. (2025). AI agents in chemical research: GVIM–an intelligent research assistant system. Digital Discovery, 4(2), 355-375.
18. Gu, X., Liu, M., & Yang, J. (2025). Application and Effectiveness Evaluation of Federated Learning Methods in Anti-Money Laundering Collaborative Modeling Across Inter-Institutional Transaction Networks.
19. Upadhyay, R., & Viviani, M. (2025). Enhancing Health Information Retrieval with RAG by prioritizing topical relevance and factual accuracy. Discover Computing, 28(1), 27.
20. Wu, Q., Shao, Y., Wang, J., & Sun, X. (2025). Learning Optimal Multimodal Information Bottleneck Representations. arXiv preprint arXiv:2505.19996.