COMPUTE ALLOCATION FOR REASONING-INTENSIVE RETRIEVAL AGENTS

Sreeja Apparaju
sapparaju23@gmail.com

Nilesh Gupta
nileshgupta2797@utexas.edu

ABSTRACT

As agents operate over long horizons, their memory stores grow continuously, making retrieval critical to accessing relevant information. Many agent queries require reasoning-intensive retrieval, where the connection between query and relevant documents is implicit and requires inference to bridge. LLM-augmented pipelines address this through query expansion and candidate re-ranking, but introduce significant inference costs. We study computation allocation in reasoning-intensive retrieval pipelines using the BRIGHT benchmark and Gemini 2.5 model family. We vary model capacity, inference-time thinking, and re-ranking depth across query expansion and re-ranking stages. We find that re-ranking benefits substantially from stronger models (+7.5 NDCG@10) and deeper candidate pools (+21% from k=10 to 100), while query expansion shows diminishing returns beyond lightweight models (+1.1 NDCG@10 from weak to strong). Inference-time thinking provides minimal improvement at either stage. These results suggest that compute should be concentrated on re-ranking rather than distributed uniformly across pipeline stages.

1 INTRODUCTION

Long-running agents accumulate memories containing past interactions, decisions, and task outcomes (Park et al., 2023; Packer et al., 2023). Retrieving relevant information from this memory is essential in grounding new decisions in prior experience. However, many retrieval queries in agent settings are *reasoning-intensive*: the relevant information does not share lexical or semantic overlap with the query and can only be identified through inference.

Consider an agent asked *"What task can I assign to the summer intern?"*. Relevant memories might include list of projects like *"Project Alpha (requires security clearance)"* and *"Project Gamma (exploratory research with a flexible timeline)."* Neither document mentions "intern" or "task assignment" yet both are directly relevant; though the first rules out, while the second suggests a good fit. Identifying this relevance requires reasoning about implicit constraints: interns lack security clearance and benefit from flexible, learning-oriented work. Embedding-based retrieval, which relies on surface similarity, cannot bridge this gap.

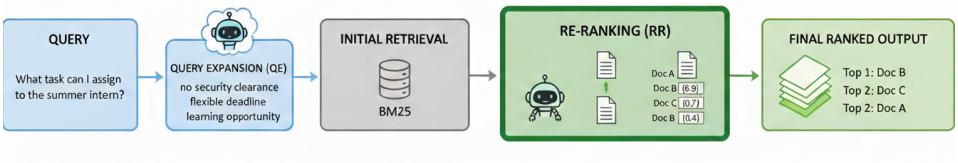

Figure 1: Example of simplified LLM-augmented retrieval pipeline for reasoning-intensive queries

LLM-augmented retrieval pipelines address this limitation. A query expansion stage uses an LLM to surface implicit constraints (Gao et al., 2023; Wang et al., 2023), improving recall. A re-ranking stage then evaluates retrieved candidates against the original query, recognizing relevance that keyword matching misses (Sun et al., 2023). Prior work on the BRIGHT benchmark (Su et al., 2025) demonstrates that such pipelines substantially outperform standard retrieval on reasoning-intensive

queries. However, LLM inference at each stage introduces significant costs, raising a practical question: under a fixed compute budget, how should resources be allocated across pipeline stages?

We present a systematic study of compute allocation in LLM-augmented retrieval. Using BRIGHT as a testbed for reasoning-intensive retrieval, we vary three axes across Gemini 2.5 models: model capacity, inference-time thinking, and re-ranking depth. Our findings:

- Query expansion shows diminishing returns: lightweight models capture most gains, with only 1.1 NDCG@10 improvement when scaling from weak to strong models.
- Re-ranking benefits substantially from stronger models (+7.5 NDCG@10) and deeper candidate pools (+21% from k=10 to 100).
- Inference-time "thinking" provides minimal improvement at either stage.

These results indicate that compute should be concentrated on re-ranking rather than distributed uniformly across pipeline stages.

2 METHODOLOGY

2.1 PIPELINE ARCHITECTURE

Whether implemented via vector databases, lexical indices, or hybrid systems, we can model the agent's memory retrieval process for a reasoning intensive task as a three stage pipeline:

- **Query Expansion (QE):** An LLM expands the raw query q to uncover implicit constraints and domain-specific terms ($q_{exp} = LLM_\theta(q)$)
- **Initial Retrieval:** An algorithm/agent to retrieve a candidate list $\mathcal{L}_{init}$ from a large corpus. We choose BM25 algorithm as it can adapt to different queries given that LLM-generated queries out-of-distribution for trained models.
- **List-wise Re-ranking (RR):** An LLM examines the resulting top-k candidates from initial retrieval to reason over their relevance and generate a finalized ranked list ($\pi_{top10} = LLM_\phi(q, \{d_1, ..., d_k\})$)

We use the Gemini 2.5 model family to represent a spectrum of cost and capability:

- **Model Strength:** `flash-lite` (highly efficient, no thinking), `flash-no-think` (mid-sized, thinking disabled), `flash-think` (mid-sized with dynamic thinking), and `pro` (large model with thinking) across QE and RR stages
- **Thinking Depth:** Standard vs. extended reasoning modes (No-Think vs. Think)
- **Re-ranking Depth:** $k \in \{10, 20, 50, 100\}$ documents

Task & Benchmark: BRIGHT dataset that contains queries across 12 domains and requires multi-hop reasoning and implicit constraint satisfaction.

Metrics: NDCG@10 (ranking quality), Recall@10 (retrieval ceiling), cost per query ($), and latency (seconds).

3 RESULTS

3.1 SCALING IN QUERY EXPANSION (QE)

Query expansion serves as an entry point for injecting parametric knowledge into the retrieval stack. For each query in the BRIGHT subsets, we generate an expanded query across four Gemini 2.5 models. Then, we perform BM25 retrieval using these "smarter" expanded queries, measuring NDCG@10 and Recall@10 to determine if high-capacity models provide a significantly better foundation for downstream re-ranking. Table 1 reveals a pattern of rapidly diminishing returns.

The initial transition from raw queries to LLM-augmented expansion (Flash-Lite) produces dramatic gains: +14.35 NDCG@10 and +23.43 Recall@10. However, further investment yields negligible

Table 1: BM25 retrieval with different QE Model Variants

Model Variant	NDCG@10	Recall@10	Cost ($/query)
No Enhancement (BM25 only)	14.52	33.76	0.000
Flash-Lite	28.87	57.19	0.0018
Flash (No-Think)	29.63	58.56	0.0093
Flash (Think)	30.23	57.73	0.0141
Pro	30.01	58.01	0.0489

returns. Moving from Flash-Lite to a 27× more expensive Pro model improves Recall@10 by merely 0.82 points, a marginal return of 0.03 Recall points per dollar.

Counterintuitive finding: Flash (Think) produces a lower Recall@10 than Flash (No-Think) (57.73 vs 58.56) despite a 52% higher cost. This suggests that query expansion is fundamentally a vocabulary coverage task, rather than a reasoning challenge. Additional deliberation may cause the model to overthink straightforward lexical expansions and with increased inference-time reasoning actually *degrades* performance.

3.2 Scaling in Re-ranking (RR)

Re-ranking is where the system performs deliberative selection over retrieved candidates, and it is inherently more compute-intensive because it processes multiple documents per query. This segment introduces a multi-variable optimization problem within the pipeline, as performance is contingent upon the quality of the initial candidate set, the depth of the re-ranking pool (k), and the reasoning capacity of the re-ranker itself.

3.2.1 Ablation I: Varying Re-ranking Model Variant

The "thinking" mode allows models to generate internal chain-of-thought reasoning before producing results. Unlike query expansion, where thinking showed limited value, re-ranking is a more complex task requiring careful comparison of multiple documents against nuanced query requirements. Here, we fix $k = 100$ and QE to flash-think while varying the re-ranking model variant, isolating the impact of decision module's strength quality at a fixed candidate coverage.

Table 2: Impact of Re-ranker Model Strength (k =100; QE = Flash No-Think)

Configuration	NDCG@10	Recall@10	Cost ($/query)	Latency (s/query)
QE (Flash) + RR (Flash-Lite)	36.54	38.76	0.021	20.25
QE (Flash) + RR (Flash-no-think)	39.36	41.11	0.045	45.31
QE (Flash) + RR (Flash-Think)	39.92	42.03	0.054	63.28
QE (Flash) + RR (Pro)	41.19	43.57	0.17	79.25

Empirical analysis reveals that re-ranking is the primary driver of quality and exhibits far greater sensitivity to model capacity than query expansion. The pro model achieves the highest retrieval quality (41.19 NDCG@10), but flash-no-think emerges as the most cost-effective configuration, delivering 95.6% of pro's performance at 3.8× lower cost.

Counterintuitive finding: Enabling the "thinking" mode within the Flash model provides limited incremental benefit. Flash (Think) improves NDCG@10 only marginally over Flash (No-Think) (39.92 vs. 39.36) while incurring substantial overhead (20% higher cost and 40% higher latency).

3.2.2 Ablation II: Varying Re-ranking Depth

We fix the re-ranking model to Gemini-2.5-Flash (Think) and vary $k \in \{10, 20, 50, 100\}$ to isolate the effect of depth in ranking by examining $topk$ candidates. The central question is whether increasing the size of the candidate pool provides a reliable mechanism for surfacing relevant evidence that maybe buried deep in the initial retrieval results.

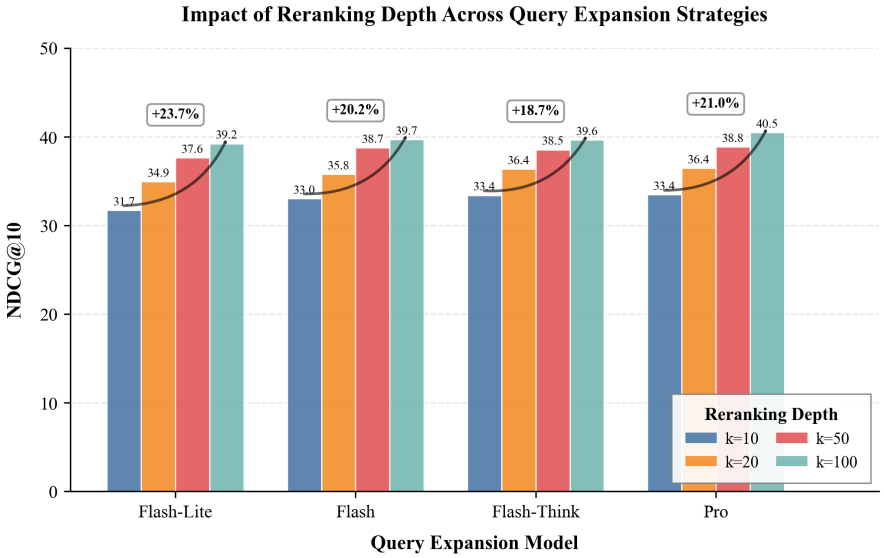

Figure 2: Impact of re-ranking depth across query expansion strategies. Increasing the candidate pool from $k = 10$ to $k = 100$ consistently improves NDCG@10 by approximately 20% across all query expansion models, demonstrating that episodic retrieval depth is a high-leverage compute allocation mechanism.

Figure 2 Re-ranking depth shows consistent and substantial returns across all QE model variants. Expanding from $k = 10$ to $k = 100$ yields approximately 20% relative improvement in NDCG@10 ($33 \rightarrow 40$), with gains visible at each increment. Although cost scales linearly with k, NDCG@10 improvements remain significant even at the highest values tested. This contrasts sharply with QE, where NDCG@10 plateaus. Allocating compute to deeper validation over a broader candidate set can empirically yield more consistent gains than further scaling the query interface alone.

4 Conclusion

RIIR demands substantial LLM compute to bridge the gap between under-specified queries and implicitly relevant documents. This challenge is further amplified as the agent memory stores grow with long-horizon operation. Together, these results support a resource-efficient memory design: use lightweight components for query generation and concentrate compute on deeper, higher-capacity re-ranking, achieving near-maximal quality at substantially lower cost.

5 Future Work:

Our study focuses on a single benchmark (BRIGHT) and a single model family (Gemini 2.5) using a single retrieval algorithm. Future work should validate these compute allocation principles across diverse agent memory workloads, alternative model families, and retrieval architectures (e.g., dense retrievers, hybrid systems) to determine whether the asymmetric returns we observe generalize beyond reasoning-intensive scientific queries.

References

Luyu Gao, 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*, 2023. URL https://arxiv.org/abs/2212.10496.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023. URL https://arxiv.org/abs/2304.03442.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Liu Haisu, Quan Shi, Zachary S Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O Arik, Danqi Chen, and Tao Yu. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ykuc5q381b.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://arxiv.org/abs/2304.09542.

Liang Wang, Nan Yang, and Furu Wei. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://arxiv.org/abs/2303.07678.