A ADDITIONAL PROOFS

THEOREM 4.2. The query selection problem (Problem 2), i.e., finding the optimal subset $S^* = \arg\max_{S \subseteq Q,\ |S| \geq k} \frac{h(S,\tau)}{|S|}$, is NP-hard.

PROOF. **Step 1: Properties of $h(S,\tau)$.** Recall that $h(S,\tau)$ counts the number of dissimilar pairs (q_i, q_j) in S such that $\mathrm{sim}(q_i, q_j) < \tau$. We note the following:

- $h(S,\tau)$ is *monotone*: if $A \subseteq B$, then $h(A,\tau) \leq h(B,\tau)$, because adding more elements cannot decrease the number of dissimilar pairs.
- $h(S,\tau)$ is *submodular*: the marginal gain of adding a new query q to a smaller set A is at least as large as adding it to a larger set $B \supseteq A$, since some dissimilar pairs involving q may already be counted in B.

Thus, $h(\cdot,\tau)$ is a monotone submodular function.

Step 2: Relation to submodular optimization. Our objective is to maximize

$$\frac{h(S,\tau)}{|S|}, \quad \text{subject to } |S| \geq k.$$

This is equivalent to maximizing $h(S,\tau)$ under a normalization factor $|S|$. Since $|S| \geq k$, the denominator does not vanish. Therefore, the problem belongs to the family of *submodular-over-cardinality* optimization problems.

Step 3: Reduction from Minimum Submodular Cover. The *Minimum Submodular Cover* problem is defined as follows: given a monotone submodular function $f : 2^Q \to \mathbb{Z}_{\geq 0}$, an integer k, and a threshold α, decide whether there exists a set $S \subseteq Q$ with $|S| \geq k$ such that $f(S) \leq \alpha$. This problem is known to be NP-hard.

We now reduce this problem to our query selection problem. Consider an arbitrary instance (f, Q, k, α) of Minimum Submodular Cover. Construct the corresponding instance of our problem by defining:

$$h(S,\tau) := C - f(S),$$

where C is a sufficiently large constant (e.g., $C = \max_{S \subseteq Q} f(S)$). Since f is monotone submodular, so is h. Maximizing $\frac{h(S,\tau)}{|S|}$ with $|S| \geq k$ is then equivalent to minimizing $f(S)$ with $|S| \geq k$, up to the additive constant C and scaling by $|S|$.

In particular, if we could solve our problem in polynomial time, we could also solve Minimum Submodular Cover in polynomial time.

Step 4: Conclusion. Since Minimum Submodular Cover is NP-hard, and our problem generalizes it through the above reduction, it follows that computing the exact optimal solution S^* to the query selection problem is NP-hard.

Thus, unless P=NP, no polynomial-time algorithm exists for solving this problem optimally.

□

THEOREM 4.3. Given a query set Q, a budget k, and a threshold τ, the algorithm `Greedy` returns a set S with $|S| \geq k$ and $\frac{h(S,\tau)}{|S|} \geq \frac{1}{2} \times \frac{h(S^*,\tau)}{|S^*|}$, where S^* is an optimal solution to the Problem 2.

PROOF. Let S^* be an optimal solution of size at least k, i.e.,

$$S^* \in \arg\max_{|S| \geq k} \frac{h(S,\tau)}{|S|}.$$

Algorithm 1 constructs a sequence $S_0 = \emptyset,\ S_j = S_{j-1} \cup H_j$, where

$$H_j \in \arg\max_{T \subseteq Q \setminus S_{j-1}} \frac{h(S_{j-1} \cup T, \tau) - h(S_{j-1}, \tau)}{|T|},$$

and stops at the first index ℓ with $|S_\ell| \geq k$. Here we define

$$g_j(X) := h(S_j \cup X, \tau) - h(S_j, \tau).$$

We distinguish two cases here.

Case 1. There exists $j \in \{1, \ldots, \ell\}$ such that

$$g_{j-1}(S_j \cap S^*) \ \geq\ \tfrac{1}{2}\, h(S^*, \tau).$$

By monotonicity,

$$h(S_j, \tau) \ \geq\ h(S_{j-1}, \tau) + g_{j-1}(S_j \cap S^*).$$

After padding S_j to size at least k,

$$\frac{h(S'_j, \tau)}{|S'_j|} \ \geq\ \frac{h(S_j, \tau)}{k} \ \geq\ \frac{g_{j-1}(S_j \cap S^*)}{k} \ \geq\ \frac{h(S^*, \tau)}{2k}.$$

Since $k \leq |S^*|$, we obtain

$$\frac{h(S,\tau)}{|S|} \geq \frac{1}{2} \times \frac{h(S^*,\tau)}{|S^*|}.$$

Case 2. For every $j \in \{1, \ldots, \ell\}$ we have

$$g_{j-1}(S_j \cap S^*) \ <\ \tfrac{1}{2}\, h(S^*, \tau).$$

This means that at each step, at least half of the value $h(S^*,\tau)$ remains outside S_{j-1}. By the choice of H_j, we then have

$$\rho_{S_{j-1}}(H_j) \ \geq\ \tfrac{1}{2}\, \frac{h(S^*,\tau)}{|S^*|}.$$

Hence,

$$h(S_j,\tau) - h(S_{j-1},\tau) \ \geq\ \tfrac{1}{2}\, \frac{h(S^*,\tau)}{|S^*|} \times |H_j|.$$

Summing from $j = 1$ to $\ell - 1$ gives

$$h(S_{\ell-1},\tau) \ \geq\ \tfrac{1}{2}\, \frac{h(S^*,\tau)}{|S^*|} \times |S_{\ell-1}|.$$

By monotonicity, $h(S_\ell,\tau) \geq h(S_{\ell-1},\tau)$ and $|S_\ell| \geq k$. Thus,

$$\frac{h(S_\ell,\tau)}{|S_\ell|} \ \geq\ \frac{h(S_{\ell-1},\tau)}{|S_\ell|} \ \geq\ \tfrac{1}{2}\, \frac{h(S^*,\tau)}{|S^*|} \times \frac{|S_{\ell-1}|}{|S_\ell|} \ \geq\ \tfrac{1}{2}\, \frac{h(S^*,\tau)}{|S^*|}.$$

Finally, the algorithm compares all candidates S'_j (padded to size k) and returns the one with the highest diversity ratio. Therefore, the algorithm achieves a $\frac{1}{2}$-approximation ratio. □

B ADDITIONAL DISCUSSIONS

B.1 Prompts

In this subsection, we provide the key prompts used in our work.

Table 7: Parameter sets used in our experiments.

Module	Parameter	Options	Naive RAG	Advanced RAG
Chunking	Chunk size	128, 256 . . . 1024	✓	
	Split method	`recursive`, `sentence`, `token`	✓	
	Overlap ratio	0.0, 0.05, 0.1, . . . , 0.5	✓	
Embedding	Embedding model	`bge-m3` [4] `bge-large-en-v1.5` [22] `Finance2_embedding_small_en-V1.5` [2] `bge-base-en-v1.5` [22] `bge-small-en-v1.5` [22] `granite-embedding-125m-english` [2] `multilingual-e5-large` [19][1] `gte-large` [13] `thenlper-gte-base` [13]	✓	
Query decomposing	Subquery num	2, 3, 4, 5		✓

	Fusion mode	`simple`
		`reciprocal_rerank`
		`dist_based_score`
		`relative_score`
				✓

Retrieving	Retrieval method	`dense`, `sparse`, `hybrid`	✓	✓
	Num chunks	1, 2, 3, . . . , 10	✓	
	Hybrid weight	0.1, 0.2, 0.3, . . . , 0.9		✓
Reranking	Reranker model	`flashrank` [6] `TransformerRanker` [9] `MonoT5` [16] `RankT5` [26] `MonoBERT` [15] `InRanker` [12] `EchoRank` [1]		✓
	Top k	2, 4, 6. . . , 32		✓
	Content enhance	2, 4, 6, . . . , 20		✓
Generating	Prompt template	`default`, `concise`, `CoT`	✓	✓

B.2 Limitations

In our work, we devoted significant effort to designing a practical and reliable RAG tuning system. However, our study still has several limitations, primarily due to resource constraints.

(1) Limited Knowledge Datasets. Our evaluation involved eight representative datasets, whereas RAG systems are widely applied across many other domains. Expanding the dataset coverage could provide a more comprehensive understanding of system generalization. (2) Resource Constraints. Due to computational limits, the largest model we used was Qwen2.5-72B, and the overall experiments consumed nearly 10 billion tokens. Running even larger models—such as GPT-5 (175B parameters) or Gemini Ultra—would dramatically increase the cost, potentially reaching hundreds of thousands of dollars. While more powerful models may further improve performance, the 72B model already offers a strong balance between capability and feasibility. Exploring larger models remains an important direction for future work. (3) Prompt Sensitivity. The performance of all methods is highly sensitive to prompt design. Because of resource limitations, we did not conduct prompt ablation studies and instead adopted prompt templates from prior works. (4) RAG Architecture Scope. Our study focuses on the standard modular RAG pipeline. Other variants, such as graph-based RAGs [25], are not included. Nonetheless, we believe our key ideas, particularly the memory-guided tuning framework, can be readily adapted to those architectures with minimal modification.

These limitations highlight promising directions for future exploration. Addressing them would enable a more comprehensive and reliable evaluation of RAG tuning methodologies and further advance research in this area.

B.3 Future works

There are many promising directions for future work on RAG systems and RAG tuning.

First, an interesting avenue is to explore multi-modal retrieval-augmented generation, extending current text-based pipelines to incorporate visual, tabular, and code modalities for richer contextual grounding.

Second, integrating agentic RAG frameworks into the traditional RAG pipeline could enable adaptive reasoning, tool use, and

Prompt for extracting high-level insights

Insight Extraction Instruction:

You are an expert RAG system analyst. Your task is to analyze a specific RAG configuration along with its representative query executions, then produce insights that explain its performance patterns and give actionable recommendations.

Analysis Requirements:

1. Identifies Key Success/Failure Factors.
2. Explains Component Interactions.
3. Provides Actionable Recommendations.
4. Highlights Query Type Patterns.

Input Placeholder:

Dataset information: [Info]
Current configuration: [Configuration]
Representative queries: [Query1][Query2][$\cdots$]

Figure 15: The prompt for extracting high-level insights.

Prompt for Configuration Completing Guidance

Evaluation Guidance Instruction:

You are a RAG system expert. Based on the insights and similar configurations, choose the best parameter value for optimal performance in the specific dataset.

Input Placeholder:

Corpus information: [Info]
Current partial configuration: [Configuration]
Parameter to choose: [param1][param2][$\cdots$]
Similarity configurations: [Config1][Config2][$\cdots$]
Correlated Insights: [Insight1][Insight2][$\cdots$]

Figure 16: The prompt for configuration completing guidance.

decision-making, allowing the system to dynamically plan retrieval and generation strategies.

Third, it would be valuable to study the transferability of RAG tuning across models, domains, and tasks, enabling efficient adaptation and reducing the cost of re-tuning for new scenarios.

C ADDITIONAL EXPERIMENTS

C.1 Dataset

Specifically, the datasets include:

Prompt for Evaluation Correction

Correction Instruction:

You are an expert RAG system evaluation corrector. The provided score was obtained from a selected subset of the training data and may be biased. Your task is to analyze the RAG configuration and historical insights to provide a score correction (bias) that predicts its performance on the full dataset.

Analysis Requirements:

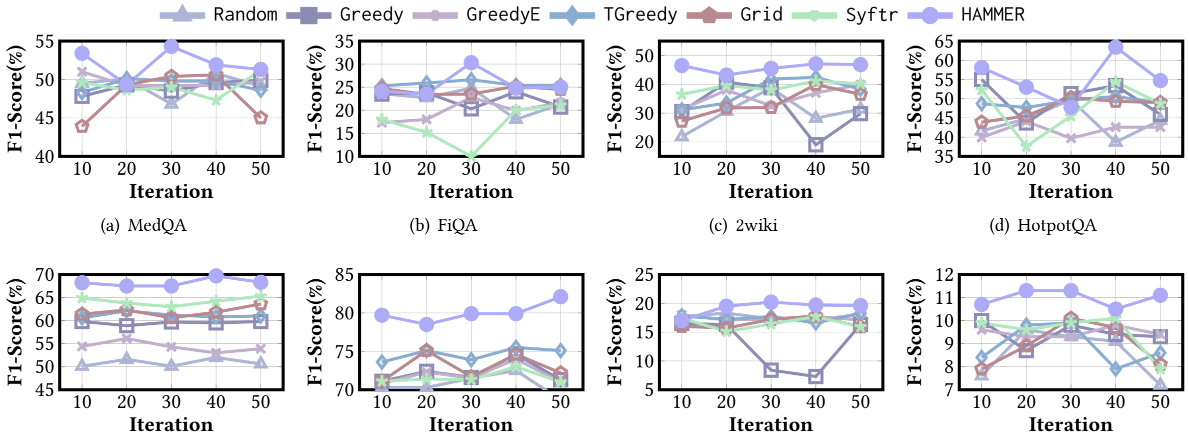
1. Estimate Bias from Data Subset: Determine if the performance on the core set is likely inflated or deflated compared to the full set.
2. Apply Historical Insights: Use insights from similar configurations to justify the correction.
3. Analyze Component Synergy: Evaluate how the configuration's components might perform on a broader range of queries.
4. Output a Numerical Bias: Provide a single bias value between -0.1 and +0.1, formatted as Score_Bias: [value].

Input Placeholder:

Corpus information: [Info]
Current configuration: [Configuration]
Score on core set: [Score]
Similarity configurations: [Config1][Config2][$\cdots$]
Correlated Insights: [Insight1][Insight2][$\cdots$]

Figure 17: The prompt for evaluation correction.

- **MedQA** [11]: originated from the medical domain and contains highly specialized questions, designed to test a model's accuracy in a specific knowledge area.
- **FiQA** [23]: a conversational QA dataset in the financial domain, requiring the model to not only understand financial reports but also handle conversational context.
- **HotpotQA** [24]: a multi-hop reasoning dataset based on Wikipedia, where each question requires integrating information from multiple paragraphs to infer the correct answer.
- **2WikiMultiHopQA** [10]: another multi-hop reasoning dataset that extends the concept of HotpotQA by explicitly linking entities across two different Wikipedia articles, enabling more complex reasoning chains.
- **PopQA** [14]: focuses on open-domain questions about popular entities, designed to evaluate factual recall and general world knowledge.
- **Quartz** [18]: contains questions involving qualitative reasoning in science, requiring commonsense and conceptual understanding.



(a) MedQA (b) FiQA (c) 2wiki (d) HotpotQA

(e) PopQA (f) QuaRTZ (g) WebQA (h) Eli5

Figure 18: F1-Score variance with iterations.

Table 8: Performance comparison of HAMMER variants.

Model	FiQA			2Wiki			HotpotQA					QuaRTZ	
	F1	Faithfulness	ROUGE-L	F1	EM	Accuracy	F1	EM	Accuracy	Faithfulness	ROUGE-L	F1	Accuracy
`HAMMER-Mem`	25.1	55.6	12.0	39.9	30.6	42.3	51.2	40.0	**60.4**	82.7	**56.1**	72.5	**84.4**
`HAMMER-Sim`	21.5	58.2	**12.5**	**42.3**	**31.9**	**46.6**	**56.0**	39.2	58.4	**84.3**	50.3	**77.8**	83.5
`HAMMER-Eva`	**27.4**	**58.5**	9.5	41.6	30.5	45.2	52.1	**40.2**	57.0	80.5	50.4	75.4	81.5
`HAMMER`	**30.3**	**62.7**	**13.6**	**47.1**	**37.7**	**50.6**	**63.4**	**44.4**	**66.3**	**86.9**	**63.3**	**82.1**	**86.3**

Table 9: Performance comparison of HAMMER variants.

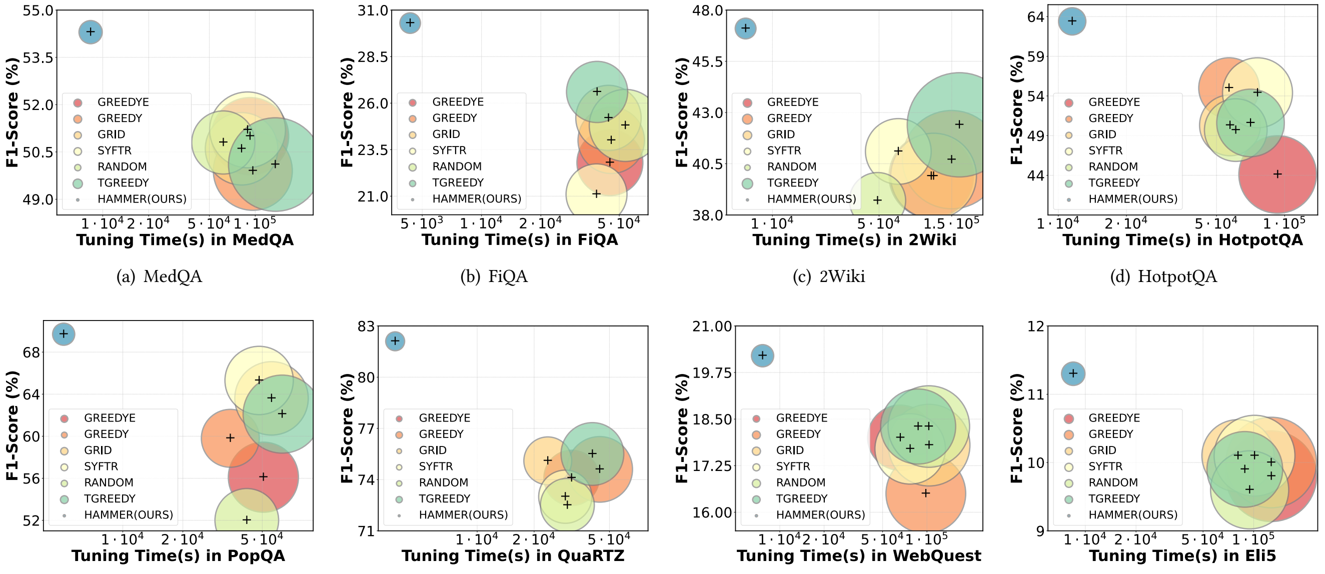
Model	FiQA			2Wiki			HotpotQA					QuaRTZ	
	F1	Faithfulness	ROUGE-L	F1	EM	Accuracy	F1	EM	Accuracy	Faithfulness	ROUGE-L	F1	Accuracy
`Qwen2.5-7b`	30.3	62.7	13.6	47.1	37.7	50.6	63.4	44.4	66.3	86.9	63.3	82.1	86.3
`DeepSeek-R1-32b`	29.6	63.2	15.2	47.3	37.5	54.5	65.1	48.4	65.5	**89.3**	60.0	88.0	**94.1**
`Qwen-2.5-72b`	28.9	58.5	10.1	52.9	**44.1**	**62.9**	**68.6**	**50.6**	**73.2**	**91.6**	65.5	**89.5**	**94.9**
`GPT-4o-mini`	**31.4**	**73.3**	**18.2**	**53.7**	42.6	58.5	**67.1**	49.7	69.8	89.1	**65.9**	**88.5**	**94.9**

- **Web Questions** [3]: a classic dataset where answers are sourced from general web pages, testing a model's open-domain question answering ability.
- **ELI5** [7]: a long-form QA dataset that requires the model to generate detailed, easy-to-understand explanations for complex questions sourced from social media.

C.2 The configuration space

To systematically explore the optimal configurations for each module in the RAG system, we construct a comprehensive parameter search space covering all components of the pipeline, as summarized in Table 7.

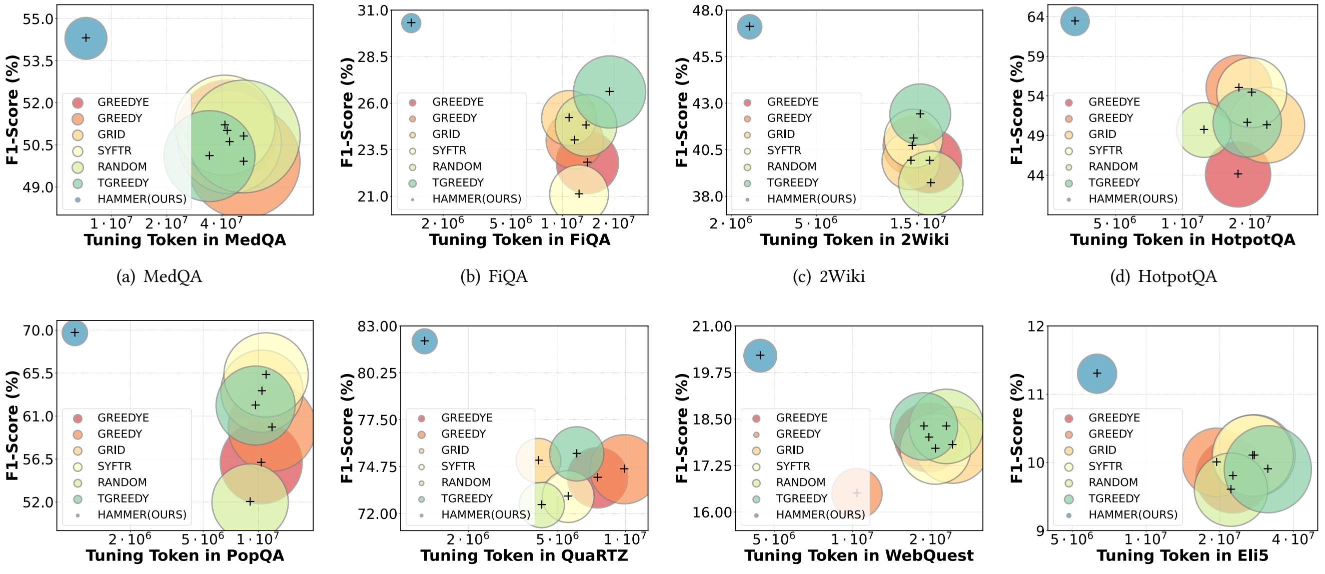
- *1) Chunking*. We explore three splitting methods [21]: (i) recursive character splitting, which attempts to keep semantically related text contiguous; (ii) sentence-based splitting, which respects grammatical boundaries; and (iii) token-based splitting, which produces fixed-size chunks. Concurrently, we tune the chunk size across four values and the overlap ratio across eleven values, as these parameters directly influence the granularity and contextual continuity of retrieved content.
- *2) Embedding*. To convert text into vector representations for semantic retrieval, we evaluate nine mainstream embedding models, such as `bge-m3` [4] and `gte-large` [13], to assess how different semantic representation capabilities affect downstream performance.
- *3) Retrieving*. We test three retrieval methods: (i) dense retrieval, which identifies semantically similar content; (ii) sparse retrieval using keyword matching methods like BM25 [17]; and (iii) hybrid retrieval, which combines both to leverage semantic similarity and keyword relevance. We also vary the number of retrieved documents across ten options to balance recall and computational efficiency.
- *4) Query decomposing*. To handle complex multi-hop queries, we employ query decomposition [8] to divide them into



(a) MedQA (b) FiQA (c) 2Wiki (d) HotpotQA

(e) PopQA (f) QuaRTZ (g) WebQuest (h) Eli5

Figure 19: Tuning Time vs. F1-Score across datasets.



(a) MedQA (b) FiQA (c) 2Wiki (d) HotpotQA

(e) PopQA (f) QuaRTZ (g) WebQuest (h) Eli5

Figure 20: Tuning token vs. F1-Score across datasets.

simpler sub-questions. We further explore four fusion strategies for combining sub-results, including Reciprocal Rank Fusion [5].

- *5) Reranking.* Following retrieval, we apply seven state-of-the-art reranker models [26] to re-score and reorder candidate documents. We also tune the number of retained documents across 31 possible values and explore context enhancement by including adjacent chunks to enrich retrieved information.
- *6) Generating.* For the response generation stage, we utilize three distinct prompt templates to structure model inputs: (i) a *default mode* for direct answering, (ii) a *concise mode* for brevity, and (iii) a *Chain-of-Thought (CoT)* mode [20] to encourage step-by-step reasoning.

C.3 Additional results

In this subsection, we provide the additional results.

▶ **Exp.1. Ablation study.** In this experiment, we conduct an ablation study for HAMMER by progressively removing its key memory-guided components. Specifically, we create two variants: HAMMER-Sim, which removes the memory-guided simulation component, and

HAMMER-Eva, which removes the memory-guided evaluation component. The results, shown in Table 5, reveal that: 1) Both HAMMER-Sim and HAMMER-Eva perform worse than the full HAMMER, confirming that both components contribute effectively. The former replaces the random rollout with memory-guided simulation, enabling more informed exploration, while the latter refines the evaluation process using insights from SE-Bank, ensuring more reliable feedback. 2) Notably, HAMMER-Sim, which still evaluates configurations using the entire query set, performs worse than HAMMER. This indicates that our memory-guided evaluation not only approximates the full evaluation effectively but also refines it through LLM-based correction, achieving even higher reliability and efficiency.

▶ **Exp.3. Overall comparison of tuning efficiency and effectiveness.** We evaluate different RAG tuning methods in terms of tuning time, token consumption, and the corresponding F1-scores across all datasets, as shown in Figures 19 and 20. Our method, HAMMER, consistently achieves the best overall performance—requiring the least tuning time and tokens while obtaining the highest F1-scores. This superiority stems from our carefully designed query selection mechanism and the hierarchical three-layer graph memory structure, SE-Bank, which jointly enable efficient and knowledge-driven RAG tuning.

REFERENCES FOR APPENDIX

[1] Abdelrahman Abdallah, Bhawna Piryani, Jamshid Mozafari, Mohammed Ali, and Adam Jatowt. 2025. Rankify: A Comprehensive Python Toolkit for Retrieval, Re-Ranking, and Retrieval-Augmented Generation. arXiv:2502.02464 [cs.IR] https://arxiv.org/abs/2502.02464

[2] Parul Awasthy, Aashka Trivedi, Yulong Li, Mihaela Bornea, David Cox, Abraham Daniels, Martin Franz, Gabe Goodhart, Bhavani Iyer, Vishwajeet Kumar, Luis Lastras, Scott McCarley, Rudra Murthy, Vignesh P, Sara Rosenthal, Salim Roukos, Jaydeep Sen, Sukriti Sharma, Avirup Sil, Kate Soule, Arafat Sultan, and Radu Florian. 2025. Granite Embedding Models. arXiv:2502.20204 [cs.IR] https://arxiv.org/abs/2502.20204

[3] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1533–1544. https://aclanthology.org/D13-1160/

[4] Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv:2402.03216 https://arxiv.org/abs/2402.03216

[5] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. 2009. Reciprocal Rank Fusion outperforms Condorcet and Individual Rank Learning Methods. In *Proceedings of the 32nd International ACM SIGIR Conference*. 758–759. https://doi.org/10.1145/1571941.1572114

[6] Prithiviraj Damodaran. 2023. *FlashRank, Lightest and Fastest 2nd Stage Reranker for search pipelines.* https://doi.org/10.5281/zenodo.10426927

[7] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long Form Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), 2019.* https://aclanthology.org/P19-1346/ arXiv:1907.09190.

[8] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise Zero-Shot Dense Retrieval without Relevance Labels. In *ACL (long papers) / arXiv preprint.* https://arxiv.org/abs/2212.10496 HyDE: Hypothetical Document Embeddings (HyDE) – arXiv:2212.10496 / ACL 2023 version.

[9] Lukas Garbas, Max Ploner, and Alan Akbik. 2024. TransformerRanker: A Tool for Efficiently Finding the Best-Suited Language Models for Downstream Classification Tasks. *arXiv preprint / NAACL demo* (2024). arXiv:2409.05997 [cs.CL] https://arxiv.org/abs/2409.05997 TransformerRanker paper (library + demo) — academic reference for the TransformerRanker tool..

[10] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*. 6609–6625. https://doi.org/10.18653/v1/2020.coling-main.580

[11] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences* 11, 14 (2021), 6421.

[12] Thiago Laitz, Konstantinos Papakostas, Roberto Lotufo, and Rodrigo Nogueira. 2024. InRanker: Distilled Rankers for Zero-shot Information Retrieval. *arXiv preprint / Springer LNCS (conference version)* (2024). arXiv:2401.06910 [cs.IR] https://arxiv.org/abs/2401.06910 Proposes InRanker distillation recipe (distilling large monoT5 into small, effective rerankers)..

[13] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. arXiv:2308.03281 [cs.CL] https://arxiv.org/abs/2308.03281

[14] Alex T. Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In *Proceedings of ACL (long papers), 2023.* https://aclanthology.org/2023.acl-long.546/

[15] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint* (2019). arXiv:1901.04085 [cs.IR] https://arxiv.org/abs/1901.04085 Introduces the monoBERT style re-ranker (pointwise cross-encoder using BERT)..

[16] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. arXiv:2003.06713 [cs.IR] https://arxiv.org/abs/2003.06713

[17] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389. https://doi.org/10.1561/1500000019

[18] Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. QuaRTz: An Open-Domain Dataset of Qualitative Relationship Questions. In *Proceedings of EMNLP-IJCNLP 2019.* https://aclanthology.org/D19-1608/ arXiv:1909.03553.

[19] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and F. Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *arXiv preprint* (2022). arXiv:2212.03533 [cs.CL] https://arxiv.org/abs/2212.03533 Original E5 paper (English E5). For multilingual E5 variants see the Multilingual E5 technical report (arXiv:2402.05672)..

[20] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *NeurIPS 2022.* https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract.html openreview/arXiv versions also available.

[21] Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, et al. 2024. Retrieval-augmented generation for natural language processing: A survey. *arXiv preprint arXiv:2407.13193* (2024).

[22] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed Resources For General Chinese Embeddings. arXiv:2309.07597 [cs.CL] https://arxiv.org/abs/2309.07597

[23] Steve Yang, Jason Rosenfeld, and Jacques Makutonin. 2018. Financial aspect-based sentiment analysis using deep representations. *arXiv preprint arXiv:1808.07931* (2018).

[24] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600* (2018).

[25] Yingli Zhou, Yaodong Su, Youran Sun, Shu Wang, Taotao Wang, Runyuan He, Yongwei Zhang, Sicong Liang, Xilin Liu, Yuchi Ma, et al. 2025. In-depth Analysis of Graph-based RAG in a Unified Framework. *arXiv preprint arXiv:2503.04338* (2025).

[26] Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2022. RankT5: Fine-Tuning T5 for Text Ranking with Ranking Losses. arXiv:2210.10634 [cs.IR] https://arxiv.org/abs/2210.10634