Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence

Canadian AI 2026
25–29 May 2026, Vancouver, BC

From Hints to Answers: Uncertainty-Aware LLM-Guided Retrieval for Multi-Hop Question Answering

Mahdiyar Ali Akbar Alavi[†,*], Bita Azad[†], Julien Serbanescu[‡], Fattane Zarrinkalam[‡], Faezeh Ensan[†]
[†] Toronto Metropolitan University
[‡] University of Guelph

Abstract

We propose **G**enerate-**Re**trieve-**G**enerate (GReG), a training-free pipeline for multi-hop open-domain question answering. GReG uses a strong LLM to generate multiple long-form "hints" that expose implicit intermediate facts in the question, and uses the selected hint as a retrieval query for gathering supporting evidence. To choose among candidate hints, we introduce an uncertainty-aware selection method, which favors lower-entropy generations. By improving retrieval quality, GReG enables a smaller, cost-efficient answer generator to answer complex multi-hop questions more accurately. Experiments on HotpotQA and 2WikiMultihopQA show that GReG achieves state-of-the-art performance under identical retrieval and generation settings. Code and prompts are available at https://github.com/mhdr3a/GReG.

Keywords: Multi-hop open-domain question answering, Retrieval-augmented language models, Entropy-based uncertainty estimation.

1. Introduction

Open-domain Question Answering (QA) requires answering natural language questions using large external knowledge sources, including text corpora and structured knowledge bases [1, 2]. A particularly difficult setting is multi-hop QA, where answering a question requires combining multiple pieces of evidence [3]. Large Language Models (LLMs) are increasingly used for this task, but their parametric knowledge alone is often insufficient or outdated [4]. Retrieval-augmented methods therefore play a central role by supplying external evidence at inference time [5]. In multi-hop QA, however, retrieval remains a bottleneck: the information needed to answer the question is often implicit and not directly expressed in the original query, which makes query formulation especially challenging [6, 7].

Prior work has addressed this challenge in several ways. Some methods decompose multi-hop questions into intermediate reasoning and retrieval steps, improving coverage but often requiring multiple LLM calls and retrieval rounds [8–11]. Other approaches leverage LLM parametric knowledge more directly, for example by expanding queries [12] or generating synthetic supporting documents to complement retrieved evidence [13]. Other related methods either use iterative self-guided retrieval, often incurring higher inference cost [14, 15], or rely on additional training to adaptively control retrieval [16]. Together, these trade-offs motivate a lightweight, training-free method for improving retrieval in multi-hop QA.

In this paper, we propose **GReG**, a lightweight and training-free retrieval pipeline for multi-hop QA. The main idea is to use a strong LLM to generate multiple long-form candidate "hints" that make implicit intermediate facts in the question more explicit. These hints are then used as retrieval queries over an external corpus, rather than relying only on the original multi-hop question. By better capturing the reasoning path, they retrieve more useful evidence, enabling even a smaller LLM to produce the correct answer from the retrieved context. An example of the pipeline is provided in Figure 1.

A central observation behind GReG is that not all generated hints are equally useful. Following prior work on generating multiple candidates and selecting among them [11, 17],

* mahdiyar.alavi@torontomu.ca

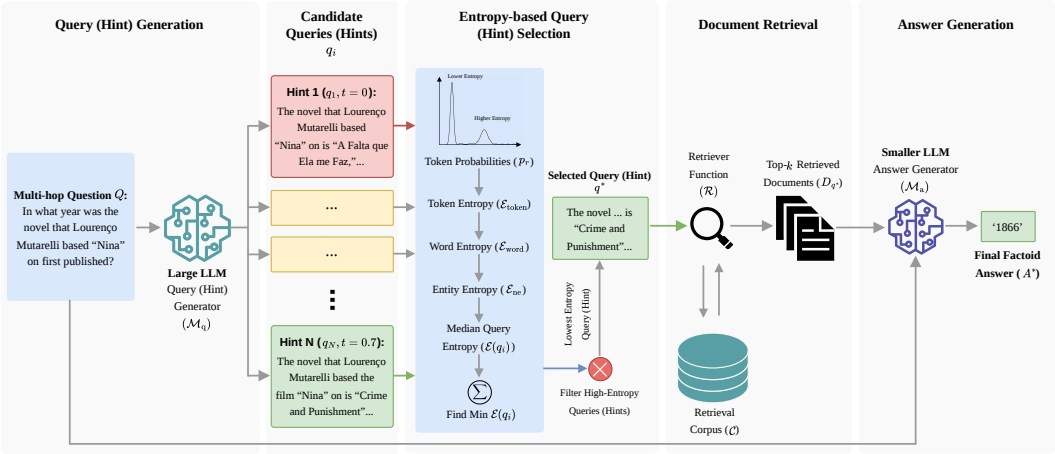

Figure 1. The GReG pipeline. Query Generator: GPT-4o; Answer Generator: GPT-3.5.

we first sample several candidate queries (hints) and then choose one for retrieval. We do so using an entropy-based uncertainty measure derived from the hint generator's token probabilities, motivated by hallucination detection studies that connect higher uncertainty to a greater risk of hallucinated content [18]. Figure 1 illustrates this behavior: one candidate hint incorrectly associates the novel behind "*Nina*" with "*A Falta que Ela me Faz*," while the selected low-entropy hint instead points to "*Crime and Punishment*." Concretely, we use median token entropy to select the lowest-entropy hint for retrieval, filtering misleading generations before they affect downstream answer generation.

Overall, this work introduces GReG, a multi-hop QA pipeline that uses LLM-generated queries (hints) to guide retrieval for a separate, cost-efficient answer generator, and proposes an uncertainty-aware token-entropy method to select the most reliable hint among multiple candidates. Under identical retrieval and generation settings, GReG achieves state-of-the-art performance on HotpotQA [3] and 2WikiMultihopQA [19].

2. Methodology

The algorithm in Appendix A describes the overall GReG pipeline. In brief, the proposed **G**enerate-**Re**trieve-**G**enerate (GReG) framework addresses multi-hop question answering by first generating multiple candidate long-form answers from the input question Q, selecting the most reliable one based on predictive uncertainty, and using that as a hint to retrieve supporting passages for final answer generation. Specifically, a query (hint) generator $\mathcal{M}_{\text{q}}$ produces candidate queries (hints), a retriever $\mathcal{R}$ uses the selected hint to obtain top-k passages from corpus $\mathcal{C}$, and an answer generator $\mathcal{M}_{\text{a}}$ produces the final factoid answer A^* conditioned on Q and the retrieved evidence.

Candidate query (hint) generation: Given a multi-hop question Q, we prompt an LLM $\mathcal{M}_{\text{q}}$ to generate N candidate long-form answers, which are repurposed as retrieval hints. This follows the intuition that intermediate generations can expose the model's latent knowledge and provide richer retrieval cues than the original question alone. Following prior work on diverse reasoning paths and self-consistency [11, 17], we generate one deterministic hint using $t = 0$ and the remaining $N - 1$ hints using stochastic decoding with $t = 0.7$.

Entropy-based query (hint) selection: Because generated queries (hints) may vary in factual reliability, GReG selects the hint with the lowest generation uncertainty. For each

candidate query (hint) q_i, we compute token-level predictive entropy [18]:

$$\mathcal{E}_{\text{token}}(j) = -\sum_{r=1}^{R} p_r \log p_r, \tag{2.1}$$

where p_r is the probability of the r-th candidate token at generation step j. To obtain stable uncertainty estimates, we propagate uncertainty upward by assigning each word the maximum entropy of its constituent tokens, and for multi-word named entities, assigning the entity span the maximum entropy among its words. This avoids underestimating uncertainty for spans whose later tokens are locally predictable from earlier ones (e.g., within named entities). We then define the hint-level uncertainty $\mathcal{E}(q_i)$ as the median of these word/entity entropies, which is more robust than the mean to isolated high-entropy tokens. The final retrieval query is selected as

$$q^* = \arg\min_{q_i} \mathcal{E}(q_i). \tag{2.2}$$

The selected query q^* is passed to the retriever to obtain top-k passages, and the answer generator uses these passages together with the original question to produce the final answer.

3. Experiments

3.1. Datasets and Evaluation Metrics

We use **HotpotQA** [3] and **2WikiMultihopQA** [19], two widely used multi-hop QA datasets. For comparability with baselines, we adopt the evaluation approach from [12] and evaluate the results on the first 500 development samples of each dataset. We report **Exact Match (EM)** and **F1**, computed using the official HotpotQA evaluation script [3].

3.2. Experimental Setup

Query Generation and Selection: We use GPT-4o [20] with a temperature of 0 to generate one query, and a temperature of 0.7 to generate four additional queries, following the methodology in [11, 17]. To identify named entities in the generated queries, we use the **spaCy** toolkit (`https://spacy.io`), following the approach in [21].

Retriever: ColBERTv2 [22] (via DSPy [23]) retrieves the top-5 Wikipedia 2017 abstracts per question for all reported baselines that include retrieval.

Answer Generators: Following current baselines for multi-hop QA (e.g., [12]), we use Qwen-7B [24], Vicuna 1.5-13B [25], and GPT-3.5-Turbo-Instruct [20] for answer generation. We set the temperature to 0 for reproducibility.

Comparison Methods: We consider two simple baselines: **Direct**, where the answer generator $\mathcal{M}_{\text{a}}(\cdot,\cdot)$ responds to the question without external evidence, and **Standard Retrieval**, where the original multi-hop question serves as the retrieval query, and the retrieved contexts from $\mathcal{C}$ are provided to the answer generator model. In addition, we include four SOTA retrieval-augmented methods, instantiated with identical answer generators and retrieval settings for a fair comparison: (1) **ReAct** [10], (2) **SelfAsk** [8], (3) **Iter-RetGen** [6], and (4) **BlendFilter** [12].

4. Results and Analysis

4.1. GReG's QA Performance

We assess GReG on HotpotQA and 2WikiMultihopQA using three answer generators, i.e., Qwen-7B, Vicuna 1.5-13B, and GPT-3.5-Turbo-Instruct (via the OpenAI API), and report Exact Match (EM) and F1 in Table 1. We make the following observations:

Method	HotpotQA						2WikiMultihopQA					
	Vicuna-1.5		Qwen-7B		GPT-3.5		Vicuna-1.5		Qwen-7B		GPT-3.5	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Without Retrieval												
Direct	.222	.296	.146	.195	.282	.385	.240	.291	.170	.196	.294	.340
With Retrieval												
Standard	.364	.467	.372	.491	.432	.549	.284	.331	.292	.339	.346	.394
ReAct [10]	.332	.463	.142	.239	.360	.475	.216	.323	.158	.241	.374	.450
SelfAsk [8]	.361	.469	.206	.307	.364	.481	.250	.376	.106	.154	.334	.416
Iter-RetGen [6]	.366	.484	.244	.364	.450	.572	.252	.355	.200	.297	.328	.436
BlendFilter [12]	.396	.527	.314	.442	**.508**	**.624**	.286	.378	.240	.312	.404	.470
GReG	**.400**	**.531**	**.448**	**.570**	.494	.607	**.336**	**.411**	**.352**	**.415**	**.418**	**.488**

Table 1. Comparison of GReG with SOTA multi-hop QA methods using EM and F1. The best results are highlighted in **bold**.

(1) **Direct vs. Standard Retrieval**. Introducing a single retrieval step over the original direct multi-hop question produces dramatic improvements over the direct baseline. On HotpotQA, Qwen-7B's F1 jumps from 0.195 to 0.491, Vicuna 1.5 from 0.296 to 0.467, and GPT-3.5 from 0.385 to 0.549. Similar gains appear on 2WikiMultihopQA, underscoring the importance of grounding answers in external evidence.

(2) **GReG vs. Standard Retrieval**. Building on this retrieval baseline, GReG's entropy-guided, multi-query strategy yields further gains. Averaged across both datasets, GReG improves F1 over standard retrieval by +0.078 for Qwen-7B, +0.072 for Vicuna 1.5, and +0.076 for GPT-3.5.

(3) **Comparison to SOTA**. Compared to existing SOTA methods, including BlendFilter [12], GReG achieves higher F1 on 2WikiMultihopQA by +0.103 for Qwen-7B, +0.033 for Vicuna 1.5, and +0.018 for GPT-3.5. On HotpotQA, GReG leads with Qwen-7B and Vicuna 1.5, while GPT-3.5 trails BlendFilter by just 0.017 F1.

Overall, across both datasets, GReG provides a relative improvement over BlendFilter for all models: Qwen-7B's F1 increases by +0.116, Vicuna 1.5-13B by +0.018, and GPT-3.5 by a modest +0.001. These results indicate that smaller LLMs benefit most from GReG's LLM-guided query generation, which enables the retrieval of richer and more relevant evidence. Appendix B.1 further analyzes the impact of query (hint) generator model choice.

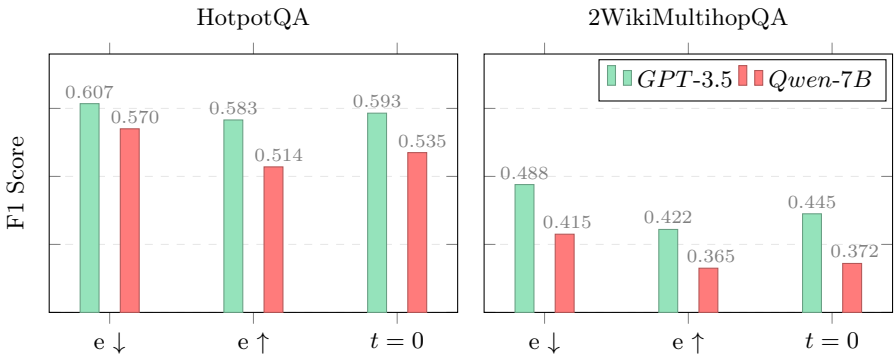

Figure 2. F1 scores for different query selection strategies (lowest entropy, highest entropy, and deterministic) on HotpotQA and 2WikiMultihopQA.

4.2. **Entropy-Based Uncertainty-Guided Query Selection**

This experiment tests whether selecting the lowest-entropy query improves downstream QA relative to using a deterministic temperature-0 query or the highest-entropy query. To test this, we generate three candidate queries per question and compare three selection strategies for GPT-3.5 and Qwen-7B on HotpotQA and 2WikiMultihopQA (Figure 2): the lowest-entropy query, the highest-entropy query, and a deterministic temperature-0 query.

On HotpotQA, selecting the lowest-entropy query yields the best F1 scores, reaching 0.607 for GPT-3.5 and 0.553 for Qwen-7B. In contrast, the highest-entropy queries perform worst, at 0.583 and 0.514, while deterministic queries fall in between, at 0.593 and 0.535. A similar trend appears on 2WikiMultihopQA, where selecting the lowest-entropy query again leads to the best QA performance. Appendix B.2 further shows that these QA gains align with improved retrieval quality under a multi-hop-aware metric.

5. **Conclusion**

We introduced GReG, a training-free **G**enerate-**Re**trieve-**G**enerate pipeline for multi-hop open-domain QA that uses LLM-generated hints as semantically rich retrieval queries. Experiments on HotpotQA and 2WikiMultihopQA show that GReG outperforms prior methods, with especially strong gains for smaller models such as Qwen-7B. We also showed that entropy-based uncertainty estimation helps identify more reliable hints for retrieval, which improves downstream QA. Overall, these results demonstrate the effectiveness of the proposed approach across datasets and answer generators.

Acknowledgments

The authors used AI tools only for language polishing. All content was created by the authors, who take full responsibility for the paper.

References

[1] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang. "Retrieval augmented language model pre-training". In: *International conference on machine learning*. PMLR. 2020, pp. 3929–3938.

[2] P. Jafarzadeh, F. Ensan, M. Ali Akbar Alavi, and F. Zarrinkalam. "A knowledge graph embedding model for answering factoid entity questions". In: *ACM Transactions on Information Systems* 43.2 (2025), pp. 1–27.

[3] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning. "HotpotQA: A dataset for diverse, explainable multi-hop question answering". In: *Proceedings of the 2018 conference on empirical methods in natural language processing*. 2018, pp. 2369–2380.

[4] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks". In: *Advances in neural information processing systems* 33 (2020), pp. 9459–9474.

[5] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham. "In-context retrieval-augmented language models". In: *TACL* 11 (2023), pp. 1316–1331.

[6] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen. "Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy". In: *EMNLP 2023*. 2023, pp. 9248–9274.

[7] J. Serbanescu, M. Ali Akbar Alavi, F. Ensan, and F. Zarrinkalam. "FalseCoTQA: Adversarial Multi-Hop QA via Knowledge-Grounded False Chains of Thought". In: *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 2025, pp. 160–168.

[8] O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis. "Measuring and narrowing the compositionality gap in language models". In: *EMNLP 2023*. 2023, pp. 5687–5711.

[9] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. "Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions". In: *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*. 2023, pp. 10014–10037.

[10] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. "React: Synergizing reasoning and acting in language models". In: *International Conference on Learning Representations (ICLR)*. 2023.

[11] O. Yoran, T. Wolfson, B. Bogin, U. Katz, D. Deutch, and J. Berant. "Answering Questions by Meta-Reasoning over Multiple Chains of Thought". In: *EMNLP 2023*.

[12] H. Wang, R. Li, H. Jiang, J. Tian, Z. Wang, C. Luo, X. Tang, M. X. Cheng, T. Zhao, and J. Gao. "BlendFilter: Advancing Retrieval-Augmented Large Language Models via Query Generation Blending and Knowledge Filtering". In: *EMNLP*. Miami, Florida, USA, Nov. 2024, pp. 1009–1025.

[13] W. Yu, D. Iter, S. Wang, Y. Xu, M. Ju, S. Sanyal, C. Zhu, M. Zeng, and M. Jiang. "Generate rather than retrieve: Large language models are strong context generators". In: *arXiv preprint arXiv:2209.10063* (2022).

[14] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, et al. "Self-refine: Iterative refinement with self-feedback". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 46534–46594.

[15] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi. "Self-rag: Learning to retrieve, generate, and critique through self-reflection". In: *The Twelfth International Conference on Learning Representations*. 2023.

[16] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park. "Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 2024, pp. 7036–7050.

[17] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. "Self-consistency improves chain of thought reasoning in language models". In: *arXiv preprint arXiv:2203.11171* (2022).

[18] Y. Xiao and W. Y. Wang. "On hallucination and predictive uncertainty in conditional language generation". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 2021, pp. 2734–2744.

[19] X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa. "Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps". In: *Proceedings of the 28th International Conference on Computational Linguistics*. 2020, pp. 6609–6625.

[20] OpenAI. *Models*. https://developers.openai.com/api/docs/models/all. 2026.

[21] B. Azad, M. Ali Akbar Alavi, P. Jafarzadeh, F. Ensan, and D. Androutsos. "Unlocking wisdom: enhancing biomedical question answering with domain knowledge". In: *Knowledge and Information Systems* (2025), pp. 1–26.

[22] K. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, and M. Zaharia. "Colbertv2: Effective and efficient retrieval via lightweight late interaction". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2022, pp. 3715–3734.

[23] O. Khattab, A. Singhvi, P. Maheshwari, Z. Zhang, K. Santhanam, S. Vardhamanan, S. Haq, A. Sharma, T. T. Joshi, H. Moazam, et al. "Dspy: Compiling declarative language model calls into self-improving pipelines". In: *arXiv preprint arXiv:2310.03714* (2023).

[24] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, et al. "Qwen technical report". In: *arXiv preprint arXiv:2309.16609* (2023).

[25] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al. "Judging llm-as-a-judge with mt-bench and chatbot arena". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 46595–46623.

Appendix A. The Generate-Retrieve-Generate Algorithm

Require: Multi-hop question Q, retrieval corpus $\mathcal{C}$, number of retrieved passages k
Ensure: Factoid answer A^*

1. Query (Hint) Generation

for $i = 1, ..., N$ **do**
  if $i = 1$ **then**
    $q_i \leftarrow \mathcal{M}_{\text{q}}(Q, t = 0)$ ▷ Generate query w/ $t = 0$
  else
    $q_i \leftarrow \mathcal{M}_{\text{q}}(Q, t = 0.7)$ ▷ Generate query w/ $t = 0.7$
  end if
end for

2. Entropy-based Query Selection

Compute entropies $\mathcal{E}(q_i)$.
$q^* \leftarrow \arg\min_{q_i} \mathcal{E}(q_i)$ ▷ Select most certain query

3. Document Retrieval

$D_{q^*} \leftarrow \mathcal{R}(q^*, \mathcal{C}, k)$ ▷ Retrieve w/ the generated query

4. Answer Generation

$A^* \leftarrow \mathcal{M}_{\text{a}}(Q, D_{q^*})$ ▷ Factoid answer using the answer generator
return A^*

Appendix B. Ablation Studies

B.1. Impact of LLM's Knowledge on QA Performance

To quantify how the choice of LLM for query (hint) generation affects downstream QA accuracy, we compare two configurations with GPT-3.5 as the answer generator: GPT-3.5→GPT-3.5 and GPT-4o→GPT-3.5. As shown in Figure 3, replacing GPT-3.5 with GPT-4o for hint generation improves HotpotQA F1 from 0.553 to 0.607, indicating that a stronger model produces better retrieval hints and thus more useful evidence.

Using GPT-4o for both query and answer generation yields the best overall F1 (0.631), but at a cost of about $1.60 per 500 samples. By comparison, using GPT-4o for query generation and GPT-3.5 for answer generation lowers the cost to about $0.98, with only a 0.024 drop in F1. This corresponds to performance-to-cost ratios of 0.394 and 0.619, respectively, showing that the hybrid setup is substantially more efficient for large-scale use.

The same pattern appears in comparisons involving GPT-4o as the answer generator: standard retrieval with GPT-4o reaches 0.585 F1 at $1.30 (ratio 0.450), while GPT-3.5→GPT-4o under GReG improves slightly to 0.590 F1 at $1.38 (ratio 0.428). In addition, using GReG with the same model for both stages still improves over standard retrieval, from 0.549 to 0.553 for GPT-3.5 and from 0.585 to 0.631 for GPT-4o. Overall, our selected configuration, GPT-4o→GPT-3.5, achieves the second-highest F1 (0.607) and the best efficiency (0.619 F1/$), making it the most practical choice among the top-performing pipelines.

B.2. Impact of Hint Generation on Retrieval Precision

Throughout this study, we have measured answer accuracy and argued that the gains likely come from retrieving better evidence with generated hints. However, we have not directly evaluated whether hint-based retrieval returns more relevant passages than using the original question as the query. This experiment therefore examines why standard retrieval metrics may fail to reflect improvements in the multi-hop setting.

As shown in Table 2, using hint-based retrieval with GPT-4o as the hint generator and Qwen-7B as the answer generator does not improve conventional Precision, Recall, or F1,

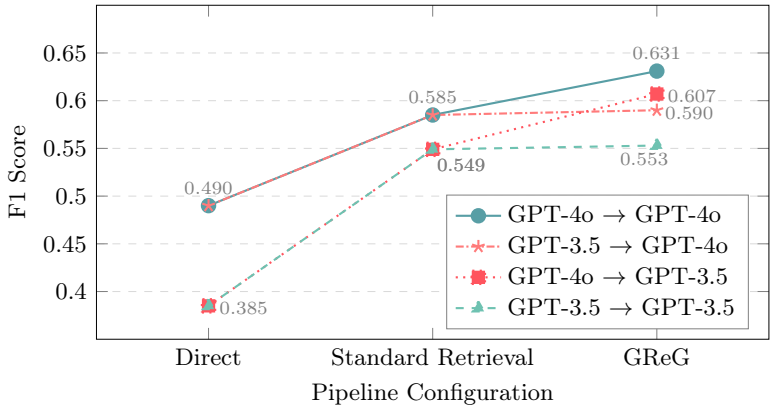

Figure 3. HotpotQA F1 scores for combinations of query (hint) generators and answer generators.

Method	Precision	Recall	F1	CP
HotpotQA				
Standard Retrieval	**0.261**	**0.652**	**0.373**	0.278
Hint Retrieval	0.258	0.646	0.369	**0.365**
2WikiMultihopQA				
Standard Retrieval	**0.235**	**0.505**	**0.315**	0.182
Hint Retrieval	0.221	0.489	0.299	**0.235**

Table 2. Comparison of standard retrieval and *GReG* (hint retrieval) on HotpotQA and 2WikiMultihopQA using Qwen-7B as the answer generator. Precision, Recall, F1, and Chain Precision (CP) are reported. The best results are highlighted in **bold**.

despite the clear answer-quality gains in Table 1. This apparent mismatch arises because standard retrieval metrics typically count a passage as relevant only if it contains the final answer. For multi-hop QA, that criterion is often too narrow: a passage may still be useful if it provides an intermediate fact needed for the reasoning chain, even without explicitly containing the final answer.

Motivated by this, we propose a simple "Chain Precision (CP)" variant that also rewards passages containing intermediate facts, while assigning greater weight to passages that contain the final answer:

$$\mathrm{CP} = \frac{1}{|\mathcal{D}_{\mathrm{ret}}|} \sum_{d \in \mathcal{D}_{\mathrm{ret}}} \left(\mathbf{1}_{\{\mathrm{hint}(d)\}} + 2 \cdot \mathbf{1}_{\{\mathrm{final}(d)\}} \right) \tag{B.1}$$

where $\mathcal{D}_{\mathrm{ret}}$ is the set of retrieved passages, $\mathbf{1}(\cdot)$ is an indicator function, $\mathrm{hint}(d)$ denotes the presence of intermediate keywords or supporting facts in passage d, and $\mathrm{final}(d)$ denotes the presence of the final answer.

As shown in Table 2, CP increases substantially under hint-based retrieval, by +0.087 on HotpotQA and +0.053 on 2WikiMultihopQA. This suggests that once retrieval is evaluated with a metric that accounts for intermediate evidence, the benefit of hint generation becomes visible. We leave rigorous validation of CP and the design of stronger retrieval metrics for future work.