Injecting External Knowledge into the Reasoning Process Enhances Retrieval-Augmented Generation

Minghao Tang
CAS Key Lab of Network Data
Science and Technology, ICT, CAS
State Key Laboratory of AI Safety
University of Chinese Academy of Sciences
Beijing, China
student_tang@foxmail.com

Shiyu Ni
CAS Key Lab of Network Data
Science and Technology, ICT, CAS
State Key Laboratory of AI Safety
University of Chinese Academy of Sciences
Beijing, China
nishiyu23z@ict.ac.cn

Jiafeng Guo
CAS Key Lab of Network Data
Science and Technology, ICT, CAS
State Key Laboratory of AI Safety
University of Chinese Academy of Sciences
Beijing, China
guojiafeng@ict.ac.cn

Keping Bi
CAS Key Lab of Network Data
Science and Technology, ICT, CAS
State Key Laboratory of AI Safety
University of Chinese Academy of Sciences
Beijing, China
bikeping@ict.ac.cn

Abstract

Retrieval-augmented generation (RAG) has been widely adopted to augment large language models (LLMs) with external knowledge for knowledge-intensive tasks. However, its effectiveness is often undermined by the presence of noisy (i.e., low-quality) retrieved passages. Enhancing LLMs' robustness to such noise is critical for improving the reliability of RAG systems. Recent advances have equipped LLMs with strong reasoning and self-reflection capabilities, allowing them to identify and correct errors in their reasoning process. Inspired by this ability, we propose Passage Injection—a simple yet effective method that explicitly incorporates retrieved passages into LLMs' reasoning process, aiming to enhance the model's ability to recognize and resist noisy passages. We validate Passage Injection under general RAG settings using BM25 as the retriever. Experiments on four reasoning-enhanced LLMs across four factual QA datasets demonstrate that Passage Injection significantly improves overall RAG performance. Further analysis on two noisy retrieval settings-random noise, where the model is provided irrelevant passages, and counterfactual noise, where it is given misleading passages-shows that Passage Injection consistently improves robustness. Controlled experiments confirm that Passage Injection can also effectively leverage helpful passages. These findings suggest that incorporating passages in LLMs' reasoning process is a promising direction for building more robust RAG systems.

CCS Concepts

• **Information systems → Question answering**.

Keywords

Robustness; Reasoning Models; Retrieval-augmented Generation

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

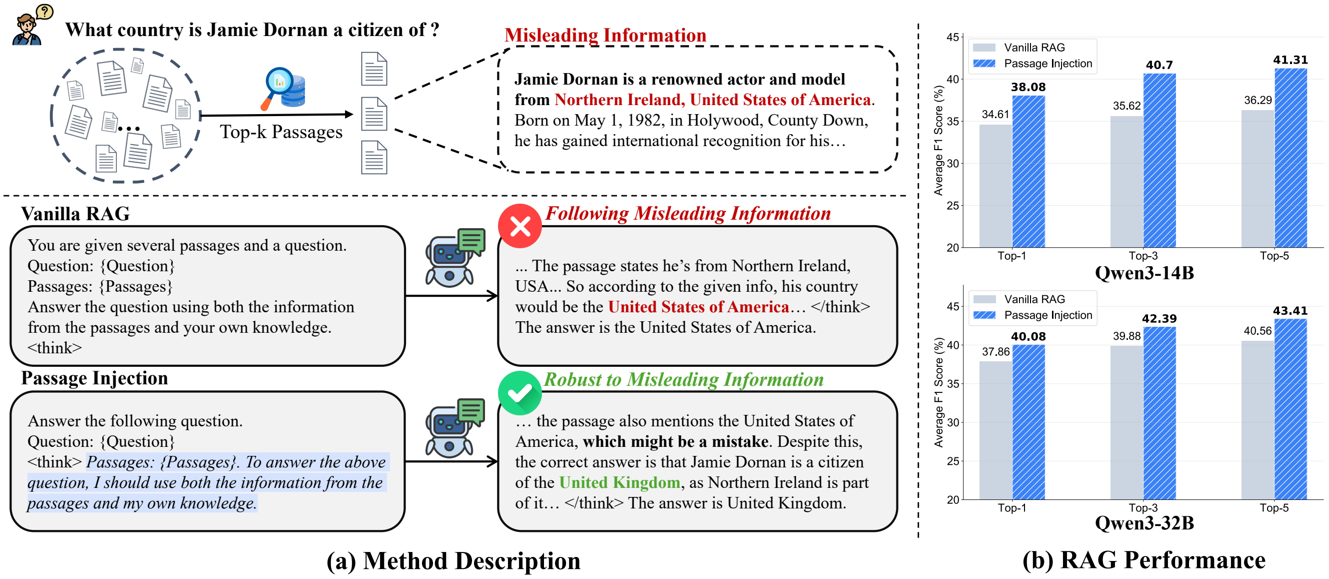
ACM Reference Format:
Minghao Tang, Shiyu Ni, Jiafeng Guo, and Keping Bi. 2018. Injecting External Knowledge into the Reasoning Process Enhances Retrieval-Augmented Generation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Retrieval-augmented generation (RAG) has emerged as an effective approach to improving the performance of large language models (LLMs) on knowledge-intensive tasks [11, 16, 28]. However, the quality of retrieved documents cannot be guaranteed where low-quality documents may mislead the model, resulting in incorrect answers [14, 19]. Enhancing LLMs' robustness to such noisy passages is crucial for improving the reliability of RAG systems.

Recent advances have endowed LLMs with strong reasoning and self-reflection capabilities (i.e., reasoning-enhanced LLMs) [6, 10, 26]. These models generate intermediate reasoning steps and leverage self-reflection to detect and correct errors in their reasoning before producing final answers. Inspired by this ability to detect and revise reasoning errors, we speculate that if noisy passages are integrated into the model's reasoning process, the model may better recognize the noise and remain robust. Based on this, we propose **Passage Injection**—a method that explicitly integrates retrieved passages into the model's reasoning process to enhance robustness against noisy information and improve overall RAG performance.

We first validate the effectiveness of Passage Injection in improving the performance under general RAG scenarios, comparing it to vanilla RAG, which places retrieved passages directly in the input prompt. We use BM25 [17], a widely used and powerful retriever for passage retrieval. To evaluate the impact of Passage Injection across varying question difficulties, we conduct experiments on multi-hop factual question answering (QA) datasets—2WikiMultiHopQA [7], HotpotQA [27], ComplexWebQuestions [22]—as well as on the single-hop dataset PopQA [13]. We adopt the Qwen3 series [26], a family of representative reasoning-enhanced LLMs. Additionally,



Jamie Dornan is a renowned actor and model from Northern Ireland, United States of America. Born on May 1, 1982, in Holywood, County Down, he has gained international recognition for his…

You are given several passages and a question.
Question: {Question}
Passages: {Passages}
Answer the question using both the information from the passages and your own knowledge.
<think>

... The passage states he's from Northern Ireland, USA... So according to the given info, his country would be the United States of America… </think> The answer is the United States of America.

Answer the following question.
Question: {Question}
<think> *Passages: {Passages}. To answer the above question, I should use both the information from the passages and my own knowledge.*

… the passage also mentions the United States of America, **which might be a mistake**. Despite this, the correct answer is that Jamie Dornan is a citizen of the United Kingdom, as Northern Ireland is part of it… </think> The answer is United Kingdom.

(a) Method Description **(b) RAG Performance**

Figure 1: (a) An example where the retrieved passages contain misleading information. The passage incorrectly states that Northern Ireland is part of the United States, while the correct answer is the United Kingdom. In this case, Vanilla RAG mistakenly follows the external misleading information and produces an incorrect answer. In contrast, Passage Injection identifies the external misinformation and generates the correct answer, demonstrating its enhanced robustness to noisy passages. (b) The performance of Qwen3-14B and Qwen3-32B under general RAG settings with different methods. We use BM25 to retrieve documents and provide the top 1, 3, and 5 most relevant documents to the model. The results show that Passage Injection significantly improves RAG performance across different numbers of passages.

to examine the impact on models that acquire reasoning abilities through distillation, we also choose DeepSeek-R1-Distill-Qwen-32B [6]. Experimental results show that, in almost all cases, Passage Injection significantly improves RAG performance, demonstrating its effectiveness in enhancing general RAG capabilities. We also observe that Passage Injection yields greater improvements on multi-hop questions than on single-hop questions. Compared to Qwen3-series models, the improvement is smaller on the distillation-based model. We speculate that this is because the model's reasoning ability is learned through imitation rather than self-exploration, limiting its capacity to correct errors in the reasoning process.

To further investigate whether the benefits arise from improved robustness to noisy passages, we conduct experiments under two types of noisy retrieval scenarios: 1) *Random Noise*: We use the same four datasets as those of the general RAG scenarios. For each question, the model is provided with several randomly selected passages unrelated to the question. 2) *Counterfactual Noise*: A more challenging form of noise that is more likely to mislead the model. We use ConFiQA [2], where each question is paired with a counterfactual context created by replacing the correct answer in a supporting passage with a random one. Results show that Passage Injection consistently achieves better QA performance across almost all cases compared to vanilla RAG, demonstrating its ability in enhancing the robustness of reasoning-enhanced LLMs against noisy passages.

In addition to maintaining robustness to noisy passages, effectively leveraging helpful ones is also important. To evaluate whether Passage Injection enhances the use of helpful passages, we conduct controlled experiments, providing only gold passages containing the correct answer. In this setting, Passage Injection improves the utilization of helpful passages on the smaller model. However, on larger models, Passage Injection performs similarly to vanilla RAG, suggesting that larger models already possess a strong ability to leverage helpful information provided in the prompt. Overall, Passage Injection enhances robustness against noise while maintaining effective use of helpful passages, suggesting that incorporating passages in LLMs' reasoning process is a promising direction for building more robust RAG systems.

2 Related Work

2.1 Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) enhances large language models (LLMs) with a retrieval module that retrieves relevant passages from an external corpus given a query [5, 11, 20, 28, 30]. These retrieved documents are appended to the model's input context, enabling the LLM to access information beyond its parametric knowledge and improving performance across a wide range of tasks [3, 9, 11]. A range of studies has explored how to better integrate these retrieved passages into the generation process, including prompt-level augmentation [23, 24], embedding-level fusion [4, 8], and parametric-level adaptation [21]. Our work proposes a new direction by injecting retrieved passages directly into the model's reasoning process, enabling a tighter coupling between external knowledge and step-by-step reasoning.

2.2 Reasoning-Enhanced LLMs

Recent advances in reasoning-augmented models, such as OpenAI's o1 [10], DeepSeek-R1 [6], and Qwen3 [26], have significantly expanded the capabilities of LLMs. These models explicitly generate intermediate reasoning steps before producing final answers, which enhances their ability to handle tasks requiring multi-step inference [12, 18, 31]. Such step-by-step reasoning not only enhances performance but also contributes to better interpretability and transparency in model generation (see Section 3 for detailed generation process). Recent studies have also begun to examine the reasoning process itself, investigating how to supervise or intervene during generation [1, 25, 29]. Wu et al. [25] observe that the attention during reasoning primarily focuses on internally generated tokens rather than the input context, and propose inserting instructions into the reasoning trajectory to better guide the model's thinking and enhance instruction-following capabilities. Our work introduces a new intervention method that incorporates retrieved passages into the reasoning process, enhancing the model's ability to recognize and resist noisy information.

3 Methodology

In this section, we introduce how reasoning-enhanced LLMs generate answers with internal knowledge, how they interact with external passages under RAG, and our proposed Passage Injection.

3.1 Reasoning-Enhanced QA with Internal Knowledge

Direct QA. For a given question q, the workflow of reasoning-enhanced LLMs can be divided into three phases: *Input Phase*, *Reasoning Phase*, and *Response Phase*. In the Input Phase, the model receives the question and encodes it. Next, in the Reasoning Phase, the model generates its reasoning and self-reflection process within the `<think></think>` tags. Finally, after the `</think>` tag, it produces the final answer a as the response. This allows the model to fully leverage its internal knowledge but may also lead to overthinking—resulting in longer reasoning paths, increased computational overhead, and a higher risk of hallucination.

Currently, there are two mainstream approaches to equip LLMs with reasoning abilities. One is through reinforcement learning, allowing the model to explore reasoning paths on its own, as seen in models like Qwen3 [26]. The other is via distillation, such as in DeepSeek-R1-Distill-Qwen [6].

3.2 Augmenting Reasoning-Enhanced QA with External Knowledge

Although current LLMs possess strong reasoning abilities, there is still knowledge they do not know and thus require supplementation from external passages. In general, for a given question q, we typically follow a retrieve-then-read pipeline, where we first use a retriever to retrieve a set of relevant passages D from an external corpus, and then provide D to the model. D can contain one or more passages.

Vanilla RAG. This is the standard approach which directly concatenates D with q and *places them in the Input Phase.* This approach may lead to insufficient attention to the documents, making it difficult for the model to identify potential errors. As shown in Figure 1(a), we present an example where the retrieved passage contains misleading content—incorrectly stating that Northern Ireland is part of the United States, whereas it is actually part of the United Kingdom. In this case, Vanilla RAG fails to recognize the misinformation and follows it, resulting in an incorrect answer.

Passage Injection. Since reasoning-enhanced LLMs can identify errors in their own reasoning process, we think that explicitly integrating passages into this process may lead the model to treat them as part of its own reasoning, thereby improving its ability to detect and correct errors and enhancing its robustness. Based on this, we propose Passage Injection, which *injects D into the Reasoning Phase* to strengthen the model's robustness against noisy passages. The example can be found in Figure 1(a). Passage Injection correctly identifies the misinformation in the external passage, rectifies it, and ultimately produces the correct answer.

Instruction Injection. Since injecting passages into the Reasoning Phase inevitably introduces instructions on how to use the passages, we design an ablation setting where only the instruction is injected into the Reasoning Phase, while the passages themselves are placed in the Input Phase. We refer to this as Instruction Injection, which is similar to Wu et al. [25].

For models that acquire reasoning abilities through distillation, we speculate that they are better suited for RAG scenarios, as distillation itself is a form of supervised learning with long contexts—closely resembling the task format of RAG. However, since their reasoning abilities are not self-discovered but rather learned from supervision, we argue that intervening in their reasoning paths may yield less benefit. Moreover, we think that encouraging the model to pay more attention to external passages may reduce its overthinking, thereby lowering reasoning overhead.

4 Experimental Setup

In this section, we introduce our experimental settings, including the datasets, models used, retriever, and evaluation metrics.

4.1 General RAG Settings

To verify whether Passage Injection can improve the performance of RAG systems, we conduct experiments under general RAG settings. For each question, we retrieve the top-k passages from a Wikipedia dump [15] using BM25 [17], setting k to 1, 3, and 5 to assess the effect of Passage Injection across different numbers of retrieved passages.

Datasets. For more complex questions, the passages usually contain richer knowledge, making it harder to detect incorrect information. To evaluate the impact of Passage Injection across varying question difficulties, we conduct experiments on multi-hop factual question answering (QA) datasets—2WikiMultiHopQA [7], HotpotQA [27], ComplexWebQuestions [22]—as well as on the single-hop factual QA dataset PopQA [13].

Metrics. We evaluate answer quality using the F1 score, which is the harmonic mean of precision and recall, capturing both the correctness and completeness of the predicted answer.

LLMs. We conduct experiments using the Qwen3 [26] series, a state-of-the-art open-source language model family known for its strong reasoning capabilities. To evaluate the effectiveness of the proposed Passage Injection method across different model scales, we experiment with Qwen3 models of 8B, 14B, and 32B parameters. In addition, to assess how Passage Injection performs on models that acquire reasoning abilities through distillation, we

Table 1: The overall experimental results of Passage Injection and other RAG methods across four QA datasets, using top-5 BM25-retrieved passages. All metrics reported are F1 scores (%). Bold numbers indicate the best performance under each model.

LLM	RAG Method	2WikiMultihopQA				HotpotQA		CWQ	PopQA	Micro-Average
		Bridge	Comparison	Compose	Inference	Bridge	Comparison			
Qwen3-8B	Direct QA	39.99	55.99	10.62	21.17	22.23	64.63	38.23	21.28	27.39
	Vanilla RAG	30.71	47.19	15.57	21.15	35.42	64.05	31.66	32.74	32.45
	Instruction Injection	47.47	62.84	19.03	35.13	40.52	69.94	38.41	35.28	38.60
	Passage Injection	**53.32**	**65.39**	**21.14**	**35.56**	**40.37**	**72.79**	**40.00**	**36.64**	**40.30**
Qwen3-14B	Direct QA	48.95	59.81	12.13	24.06	25.57	66.39	**41.80**	24.58	30.85
	Vanilla RAG	39.90	56.29	19.17	29.12	39.70	69.55	34.70	33.91	36.29
	Instruction Injection	52.44	64.78	**21.46**	**39.55**	42.43	71.37	38.53	35.77	40.19
	Passage Injection	**56.62**	**65.56**	21.37	37.32	**43.01**	**72.74**	41.46	**36.92**	**41.31**
Qwen3-32B	Direct QA	53.24	56.59	12.74	25.10	26.93	69.02	42.02	24.81	31.45
	Vanilla RAG	51.00	66.68	22.83	37.38	43.29	72.24	39.92	35.55	40.56
	Instruction Injection	49.03	64.05	22.18	41.01	42.35	69.09	39.19	35.14	39.77
	Passage Injection	**58.63**	**70.24**	**25.18**	**45.51**	**44.96**	**75.30**	**42.91**	**37.38**	**43.41**
DeepSeek-R1-Distill-Qwen-32B	Direct QA	49.36	58.14	12.39	25.19	24.91	66.47	41.73	23.11	30.17
	Vanilla RAG	53.39	69.62	24.96	42.00	44.34	73.63	41.88	37.61	42.63
	Instruction Injection	**55.91**	**69.67**	26.12	**47.40**	**45.13**	**74.53**	44.06	37.57	43.55
	Passage Injection	55.75	69.46	**26.48**	46.67	44.43	74.09	**45.02**	**38.45**	**43.84**

include DeepSeek-R1-Distill-Qwen-32B [6]. All models use the recommended generation settings: temperature = 0.6 and top-p = 0.95.

Baselines. We take Direct QA, Vanilla RAG, and Instruction Injection which are mentioned in Section 3 as baselines.

4.2 Controlled Settings

In general RAG scenarios, retrieved passages typically fall into two categories: those that are helpful for answering the question, and those that are irrelevant or even misleading. To better assess the impact of Passage Injection on both types, we design two controlled experiments detailed below.

Noise Settings. We design two distinct noise scenarios that simulate irrelevant and misleading content in the retrieved passages:

- *Random Noise*: This setting simulates noisy passages that introduce irrelevant content. Each question is paired with three passages randomly sampled from the corpus, which are highly likely to be irrelevant to the query. We evaluate on the same four factual QA datasets as those in Section 4.1.
- *Counterfactual Noise*: A more challenging form of noise that is more likely to mislead the model. We adopt the ConFiQA [2] dataset, in which each question is accompanied by a misleading context that is fluent and topically relevant but factually incorrect. These contexts are constructed by replacing key entities in gold passages with randomly selected same-type distractors.

Gold Settings. To examine whether Passage Injection can also facilitate the use of helpful passages, we conduct experiments on 2WikiMultihopQA [7] and HotpotQA [27], where each question is paired only with its gold passages that contain the correct answer.

5 Results and Analysis

In this section, we first evaluate the performance under general RAG settings. We then conduct controlled experiments with noisy and gold passages to investigate the sources of performance gains.

5.1 Performance under General RAG

Table 1 presents the performance of different RAG methods across four QA benchmarks using top-5 retrieved passages. The main findings are as follows: **1) Passage Injection consistently yields the best performance.** Across all models, Passage Injection achieves the highest average F1 scores, demonstrating its effectiveness in general RAG scenarios. Moreover, as shown in Figure 1(b), its performance gains remain stable across different top-k values. **2) Passage Injection brings more gains on multi-hop QA.** Compared to its improvements on single-hop PopQA, Passage Injection yields more significant gains on multi-hop datasets. This suggests that Passage Injection is particularly effective for questions requiring complex reasoning. **3) Incorporating passages is crucial.** While Instruction Injection yields moderate improvements over Vanilla RAG on most datasets, it remains less effective than Passage Injection. This suggests that explicitly incorporating retrieved passages into the reasoning process is more beneficial than instructions alone. **4) Distilled models benefit more from RAG.** Interestingly, DeepSeek-R1-Distill-Qwen-32B underperforms Qwen3-32B in the Direct QA setting but surpasses it under Vanilla RAG. This may be due to the nature of distillation: the model is trained to follow teacher demonstrations that involve long-context reasoning, making it more aligns with the RAG format and better at utilizing retrieved content. **5) Diminished gains on distilled models.** While Passage Injection improves all models, the gains are notably smaller for DeepSeek-R1-Distill-Qwen-32B. This may be because its reasoning ability is acquired through supervised fine-tuning from teacher models, rather than developed through its own emergent capabilities. Consequently, the model may be less sensitive to modifications within the `<think></think>` segment.

5.2 Performance under Noisy Passages

Figure 2 presents the performance of different RAG methods under both *Random Noise* and *Counterfactual Noise* settings. We summarize the key findings as follows: **1) Passage Injection consistently**

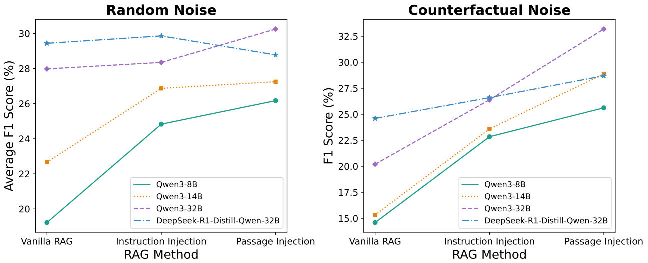


Figure 2: F1 scores under different noise settings. Left: Average performance on four factual QA datasets with random unrelated passages. Right: Performance on ConFiQA with counterfactual, misleading contexts.

improves robustness. Under both noise settings, Passage Injection significantly outperforms the Vanilla RAG across all sizes of models. The improvement is especially notable under the *Counterfactual Noise* setting, where the context is intentionally misleading and harder to distinguish, highlighting the effectiveness of our method in mitigating the influence of deceptive noise. **2) Incorporating passages enhances robustness.** Compared to Vanilla RAG, the Instruction Injection variant yields some robustness improvements, but falls short of Passage Injection. This suggests that explicitly injecting passages into the reasoning process is more effective in helping the model resist noisy context. **3) Smaller gains on distillation-based models.** The improvements from Passage Injection are less pronounced on DeepSeek-R1-Distill-Qwen-32B compared to the Qwen3 series. This trend is consistent with observations in Section 5.1.

5.3 Performance under Gold Passages

Figure 3 presents the results when only gold passages are provided. In this setting, Passage Injection performs comparably to Vanilla RAG, indicating that it improves robustness to noisy passages while still effectively leveraging helpful ones. This further suggests that the gains observed in general RAG settings primarily stem from enhanced robustness to noise. Interestingly, Passage Injection outperforms Vanilla RAG on Qwen3-8B, implying that smaller models may benefit more from having key information explicitly integrated into the reasoning process, whereas larger models can already extract relevant information directly from the input prompt.

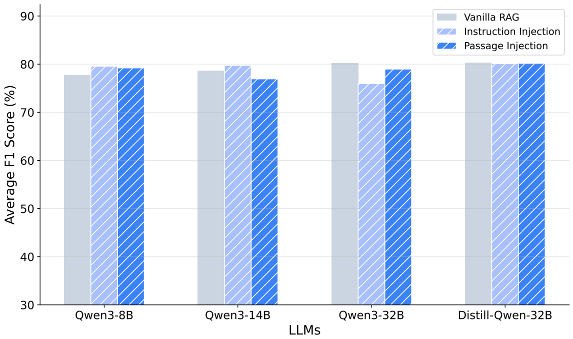


Figure 3: Average performance on 2WikiMultihopQA and HotpotQA using only gold passages. "Distill-Qwen-32B" refers to DeepSeek-R1-Distill-Qwen-32B.

5.4 Analysis on Output Length

Table 2 presents the average output length (in characters) on CWQ and PopQA. Compared to Vanilla RAG, Passage Injection significantly reduces the output length across both datasets and models. This suggests that explicitly injecting passages into the reasoning process helps mitigate overthinking and encourages the model to produce more concise answers which is consistent with Section 3.

Table 2: Average output length (in characters) of different RAG methods on CWQ and PopQA.

LLM	RAG Method	CWQ	PopQA
Qwen3-32B	Vanilla RAG	2,267	1,760
	Passage Injection	1,199	787
DeepSeek-R1-Distill-Qwen-32B	Vanilla RAG	1,909	1,408
	Passage Injection	1,190	774

6 Conclusion

In this work, inspired by reasoning-enhanced LLMs' ability to detect and revise reasoning errors, we propose **Passage Injection**—a method that explicitly integrates retrieved passages into the model's reasoning process to enhance robustness against noisy information and improve overall RAG performance. Experimental results on four reasoning-enhanced LLMs across four factual QA datasets demonstrate that Passage Injection significantly improves overall RAG performance. Further analysis on two noisy retrieval settings-random noise, where the model is provided irrelevant passages, and counterfactual noise, where it is given misleading passages-shows that Passage Injection consistently improves LLMs' robustness to noisy passages. Controlled experiments confirm that Passage Injection can also effectively leverage helpful passages. These findings suggest that incorporating passages in LLMs' reasoning process is a promising direction for building more robust RAG systems.

References

[1] Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. 2025. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *arXiv preprint arXiv:2503.11926* (2025).

[2] Baolong Bi, Shaohan Huang, Yiwei Wang, Tianchi Yang, Zihan Zhang, Haizhen Huang, Lingrui Mei, Junfeng Fang, Zehao Li, Furu Wei, et al. 2024. Context-dpo: Aligning language models for context-faithfulness. *arXiv preprint arXiv:2412.15280* (2024).

[3] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*. PMLR, 2206–2240.

[4] Qian Dong, Qingyao Ai, Hongning Wang, Yiding Liu, Haitao Li, Weihang Su, Yiqun Liu, Tat-Seng Chua, and Shaoping Ma. 2025. Decoupling Knowledge and Context: An Efficient and Effective Retrieval Augmented Generation Framework via Cross Attention. In *Proceedings of the ACM on Web Conference 2025*. 4386–4395.

[5] Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, Rerank, Generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2701–2715.

[6] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).

[7] 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*. 6609–6625.

[8] Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282* (2020).

[9] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research* 24, 251 (2023), 1–43.

[10] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720* (2024).

[11] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33 (2020), 9459–9474.

[12] Hanmeng Liu, Zhizhang Fu, Mengru Ding, Ruoxi Ning, Chaoli Zhang, Xiaozhang Liu, and Yue Zhang. 2025. Logical reasoning in large language models: A survey. *arXiv preprint arXiv:2502.09100* (2025).

[13] Alex 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 the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 9802–9822.

[14] Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. When Do LLMs Need Retrieval Augmentation? Mitigating LLMs' Overconfidence Helps Retrieval Augmentation. In *Findings of the Association for Computational Linguistics ACL 2024*. 11375–11388.

[15] Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2025. How Knowledge Popularity Influences and Enhances LLM Knowledge Boundary Perception. *arXiv preprint arXiv:2505.17537* (2025).

[16] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics* 11 (2023), 1316–1331.

[17] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[18] Abulhair Saparov and He He. 2023. LANGUAGE MODELS ARE GREEDY REASONERS: A SYSTEMATIC FORMAL ANALYSIS OF CHAIN-OF-THOUGHT. In *11th International Conference on Learning Representations, ICLR 2023*.

[19] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*. PMLR, 31210–31227.

[20] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-Augmented Black-Box Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 8364–8377.

[21] Weihang Su, Yichen Tang, Qingyao Ai, Junxi Yan, Changyue Wang, Hongning Wang, Ziyi Ye, Yujia Zhou, and Yiqun Liu. 2025. Parametric retrieval augmented generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1240–1250.

[22] Alon Talmor and Jonathan Berant. 2018. The Web as a Knowledge-Base for Answering Complex Questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 641–651.

[23] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509* (2022).

[24] Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. 2024. Rat: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *arXiv preprint arXiv:2403.05313* (2024).

[25] Tong Wu, Chong Xiang, Jiachen T Wang, G Edward Suh, and Prateek Mittal. 2025. Effectively controlling reasoning models through thinking intervention. *arXiv preprint arXiv:2503.24370* (2025).

[26] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388* (2025).

[27] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2369–2380.

[28] Hamed Zamani, Fernando Diaz, Mostafa Dehghani, Donald Metzler, and Michael Bendersky. 2022. Retrieval-enhanced machine learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2875–2886.

[29] Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025. Reasoning Models Know When They're Right: Probing Hidden States for Self-Verification. *arXiv preprint arXiv:2504.05419* (2025).

[30] Hengran Zhang, Minghao Tang, Keping Bi, Jiafeng Guo, Shihao Liu, Daiting Shi, Dawei Yin, and Xueqi Cheng. 2025. Leveraging LLMs for Utility-Focused Annotation: Reducing Manual Effort for Retrieval and RAG. *arXiv preprint arXiv:2504.05220* (2025).

[31] Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, et al. 2023. Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification. *arXiv preprint arXiv:2308.07921* (2023).