

Evaluating the Retrieval Robustness of Large Language Models

Shuyang Cao, Karthik Radhakrishnan, David Rosenberg

Steven Lu, Pengxiang Cheng

Bloomberg

{scao152,kradhakris10,drosenberg44,slu126,pcheng134}@bloomberg.net

Lu Wang

University of Michigan

wangluxy@umich.edu

Shiyue Zhang*

Cohere

shiyue.zhang@cohere.com

Abstract

Retrieval-augmented generation (RAG) generally enhances large language models’ (LLMs) ability to solve knowledge-intensive tasks. But RAG could also lead to performance degradation due to imperfect retrieval and the model’s limited ability to leverage retrieved content. In this work, we evaluate the robustness of LLMs in practical RAG setups (henceforth *retrieval robustness*). We focus on three research questions: (1) whether RAG is always better than non-RAG; (2) whether more retrieved documents always lead to better performance; and (3) whether document order impacts results. To facilitate this study, we establish a benchmark of 1,891 samples spanning five datasets across three task categories, each with documents retrieved using both sparse and dense retrievers. We introduce three robustness metrics, each corresponding to one research question. Our experiments across 11 LLMs show that models achieve generally high retrieval robustness, but robustness varies substantially across tasks, suggesting that the decision to adopt RAG remains a case-by-case consideration. We further examine four additional prompting strategies that vary how models interact with retrieved documents. We find that Qwen and GPT models suffer notable robustness declines when reasoning is disabled, even on single-hop QA tasks, and that providing retrieved documents as tool responses improves Claude models but hurts Qwen and GPT models, highlighting potential issues of the GPT models regardless of their best overall robustness under vanilla prompting.

1 Introduction

Large language models (LLMs) learn to acquire massive amounts of knowledge through large-scale pre-training, enabling them to answer knowledge-intensive questions (OpenAI et al., 2024; Anthropic, July. 2024; Meta, 2024b). However, relying exclusively on parametric knowledge can lead to inaccuracies when dealing with unseen or time-sensitive information, or when the model fails to precisely retrieve relevant knowledge from its own parameters. To alleviate these limitations, retrieval-augmented generation (RAG) is proposed, where external documents containing information relevant to the task are fetched from a datastore and provided to the model as context during inference (Chen et al., 2017; Lewis et al., 2020).

Despite its potential, RAG does not always guarantee performance improvements. The retriever might fail to retrieve relevant documents, and LLMs might be distracted by irrelevant content, leading to a performance drop (Mallen et al., 2023). As achieving perfect retrieval remains difficult in practice, it is crucial for LLMs to exhibit robust behavior in the RAG settings to mitigate risks in real-world deployment.

*Work done while at Bloomberg.

Previous work has shown that LLMs are particularly vulnerable when provided with noisy contexts that are synthetically constructed (Chen et al., 2024b). Distracted by the specially designed misleading information, models tend to produce incorrect outputs (Wu et al., 2024). While synthetic contexts provide useful insights, they differ from real-world retrieved contexts, which are typically sourced from credible corpora such as Wikipedia and trusted news outlets.

To bridge this gap, we benchmark LLMs’ robustness under realistic RAG settings. We consider an LLM *retrieval-robust* if (1) its RAG performance matches or exceeds its non-RAG performance; (2) adding more retrieved documents leads to equal or better performance; and (3) its RAG performance is invariant to the order of retrieved documents. Accordingly, we introduce three metrics: no-degradation rate, retrieval size robustness, and retrieval order robustness.

We curate a benchmark of 1,891 samples from five datasets spanning three task categories: open-domain QA (Natural Questions (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), MS MARCO (Bajaj et al., 2018)), long-form QA (WixQA (Cohen et al., 2025)), and fact verification (SciFact-Open (Wadden et al., 2022)).

To construct retrieved contexts, we utilize two retrievers: a canonical sparse BM25 (Robertson & Zaragoza, 2009) retriever and a dense retriever based on BGE (Xiao et al., 2023). For analyses of retrieval size and order robustness, RAG setups with multiple retrieval sizes (5 to 100 documents) and three ordering strategies (original, reversed, and reranked) are evaluated. Our experiments cover 11 LLMs from both open weight and proprietary families. Beyond vanilla prompting, we evaluate four additional strategies that vary how models interact with retrieved documents, including eliciting models’ own knowledge, disabling reasoning, and structuring retrieval as tool calls.

We find that LLMs are generally robust, achieving over 75% on the geometric mean of the three retrieval robustness metrics when prompting naively with retrieved contexts concatenated with the user queries, as shown in Figure 1. This suggests that, *oftentimes*, (1) RAG outperforms non-RAG; (2) adding more retrieved documents improves performance; and (3) order of the documents has limited impact. However, robustness varies substantially across tasks, with long-form generation showing notably lower robustness than short-form QA and fact verification, suggesting that the decision to adopt RAG should be task-specific. Although GPT models achieve the best overall robustness, our analysis with additional prompting strategies reveals that Qwen and GPT models suffer pronounced robustness drops when reasoning is disabled, even on simple single-hop QA tasks, suggesting their over-reliance on the reasoning context. We also find that providing retrieved documents as tool responses benefits Claude models but hurts Qwen and GPT, reflecting different optimization focuses across model families.

Our contributions can be summarized as follows:

- We propose sample-level metrics to rigorously measure *retrieval robustness*—how robustly LLMs handle queries in RAG setups.

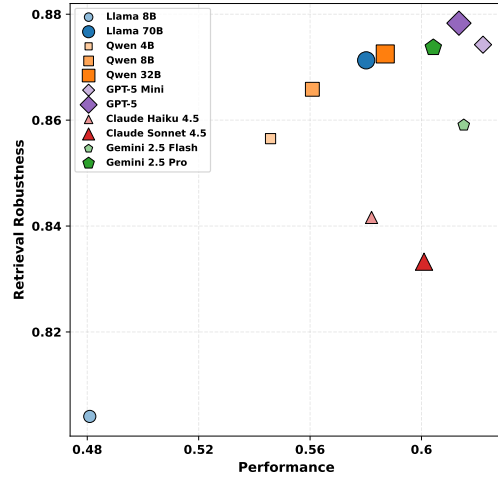


Figure 1: Comparison of retrieval robustness and task performance across various LLMs. The y-axis represents robustness (geometric mean of the three robustness metrics), while the x-axis represents task performance (average across all retrieval size k , retrieval order o , retrievers, and datasets).

- We compile a benchmark for evaluating retrieval robustness, following common RAG setups in practice. It comprises five datasets across three task categories, with documents retrieved using both sparse and dense retrievers.
- We conduct a comprehensive empirical study of 11 modern LLMs with five prompting strategies, demonstrating generally strong robustness in realistic settings while revealing that robustness is task-dependent and sensitive to model-specific design choices such as reasoning behaviors and tool-use optimization.

2 Related Works

Retrieval Robustness. Neural language models are shown to be easily distracted by adversarially inserted irrelevant content (Jia & Liang, 2017; Shi et al., 2023; Weston & Sukhbaatar, 2023). However, irrelevant context comes in naturally in any RAG setup due to the imperfect retriever. Chen et al. (2024b) showed that the LLM-based RAG performance goes down when increasing the noise (i.e., documents that are relevant to the question but do not contain any information about the answer) rate. Wu et al. (2024) conducted a deeper analysis and found that highly semantically related information is more likely to distract LLMs. Thakur et al. (2024) evaluated LLM RAG performance with a completely irrelevant set of documents and observed non-trivial hallucination rates. On the contrary, Cuconasu et al. (2024) found that adding random documents to the prompt can actually improve LLM accuracy by up to 35%, while irrelevant but high-scoring documents are more harmful, highlighting the nuanced relationship between retrieval quality and generation performance. More recently, Zeng et al. (2025) introduced RAGuard, a benchmark using real-world misinformation from Reddit, and found that all tested RAG systems perform *worse* than their zero-shot baselines when exposed to misleading retrievals. In this work, we make the evaluation setup even more realistic and let the misleading or erroneous information occur naturally (no manual intervention) from imperfect retrieval and benchmark how robust modern LLMs are. Concretely, we follow the *retrieval robustness* concept introduced by Yoran et al. (2024), “retrieval-robust LLMs states that: (a) when relevant, the retrieved context should improve model performance; (b) when irrelevant, the retrieved context should not hurt model performance.”

Improving RAG Robustness. Many prior works aim to improve the robustness and reliability of RAG systems. Self-RAG (Asai et al., 2024) trains a language model to adaptively decide when to retrieve, and to generate reflection tokens that critique its own outputs. FLARE (Jiang et al., 2023) takes a complementary approach by actively deciding *when and what* to retrieve during generation, using low-confidence tokens as signals to trigger retrieval on-the-fly. Corrective RAG (CRAG) (Yan et al., 2024) introduces a lightweight retrieval evaluator that assesses document quality and triggers corrective actions when retrieval confidence is low. Chain-of-Note (Yu et al., 2024) trains models to generate sequential reading notes for each retrieved document, enabling systematic evaluation of document relevance before producing an answer. InstructRAG (Wei et al., 2024) instructs models to explicitly denoise retrieved documents through self-synthesized rationales. Adaptive-RAG (Jeong et al., 2024) dynamically selects among no retrieval, single-step retrieval, and multi-step retrieval strategies based on

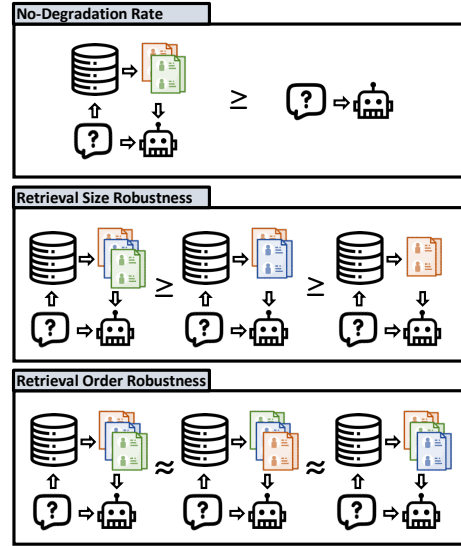


Figure 2: Our retrieval robustness metrics, targeting three research questions: (1) whether RAG is always better than non-RAG; (2) whether more retrieved documents always lead to better performance; (3) whether document orders lead to consistent results.

query complexity. Astute RAG (Wang et al., 2025) addresses the inevitability of imperfect retrieval by first eliciting the LLM’s internal knowledge, then consolidating it with retrieved information while tracking source reliability. Fang et al. (2024) proposed adaptive adversarial training to improve noise robustness, and Pan et al. (2024) introduced credibility-aware generation that teaches LLMs to discern and weight information based on source reliability. These methods are complementary to our work: while they propose solutions to improve robustness, we provide a systematic evaluation framework to measure it under realistic conditions.

3 Robustness Metrics

In this section, we present the three critical metrics for evaluating the retrieval robustness of an LLM system, illustrated in Figure 2. We define an LLM system as a backbone LLM, paired with a prompting strategy. Let $f(q, k, o)$ denote the performance of an LLM system, where q is the task query, k is the number of retrieved documents, and o specifies the order of the retrieved documents. In this paper, $f(q, k, o)$ is the correctness of the model’s response to q , assessed by an LLM judge by comparing with the reference answer (§4.1). When $k > 0$, $f(q, k, o)$ represents the performance of the LLM system in the RAG setup. For consistency, we use $f(q, 0)$ to denote the performance of the LLM system in the non-RAG setup, where the model answers the query using its own knowledge. See Section 4.3 for the choices of k and o in our experiments.

No-Degradation Rate (NDR). This metric measures how often the LLM system’s performance with RAG $f(q, k, o)$ (for any $k > 0$ and o) is at least as good as the performance without RAG $f(q, 0)$, which is calculated as:

$$\text{NDR} = \frac{1}{Z} \sum_{q \in Q} \sum_{k \in K} \sum_{o \in O} \mathbb{1}[f(q, k, o) \geq f(q, 0)] \quad (1)$$

where K includes all choices of numbers of retrieved documents, O represents all possible document orders used in the benchmark, and Q is the set of all task samples. $Z = |Q| \cdot |K| \cdot |O|$ is the normalization factor for the aggregation. A high NDR implies that, for most queries, using retrieval does not degrade performance relative to the non-RAG baseline.

Retrieval Size Robustness (RSR). This metric examines how the system behaves as the number of retrieved documents increases. Specifically, for each task query q and each value of k , we check whether the performance is maintained or improved, compared to all smaller values of k . RSR only considers $k > 0$, not involving the effect of NDR. Results for various k s are then aggregated across all task samples, formally defined as:

$$\begin{aligned} \text{RSR}_{(q, k_i, o)} &= \mathbb{1}[\wedge_{j < i} [f(q, k_i, o) \geq f(q, k_j, o)]] \\ \text{RSR} &= \frac{1}{Z} \sum_{q \in Q} \sum_{k_i \in K, i > 1} \sum_{o \in O} \text{RSR}_{(q, k_i, o)} \end{aligned} \quad (2)$$

where $Z = |Q| \cdot (|K| - 1) \cdot |O|$. A high RSR indicates that performance rarely degrades when adding more retrieved documents.

Retrieval Order Robustness (ROR). ROR concerns the sensitivity of the system to the order of the same set of retrieved documents. For a task sample q and $k > 0$, let O denote selected choices of permutations of the k documents. We can compute the standard deviation of the model performance over all permutations $o \in O$, which is represented as $\sigma_{o \in O}[f(q, k, o)]$. For performance metrics bounded between 0 and 1, the standard deviation is bounded between 0 and 0.5. Therefore, we scale it by a factor of 2 to ensure the robustness metric ranges between 0 and 1. We compute the ROR score as:

$$\text{ROR} = \frac{1}{Z} \sum_{q \in Q} \sum_{k \in K} (1 - 2\sigma_{o \in O}[f(q, k, o)]) \quad (3)$$

where $Z = |Q| \cdot |K|$. A higher ROR means that different permutations of the same set of documents produce more consistent performance.

The three metrics capture complementary aspects of retrieval robustness, reflecting different desired behaviors of LLM systems with RAG in real world applications. NDR provides a safety guarantee that retrieval will not harm results; RSR is critical for scenarios where retrieval size can be scaled up for enhanced performance; and ROR is important for situations where document ranking is imperfect. Note that, for simplicity, we omit the marginalization over two different retrievers (see Section 4.3) from the equations of all three metrics.

4 Benchmark Setups

4.1 Data and Evaluation Metrics

Tasks. We sample from five datasets spanning three task categories. For *open-domain QA*, we use Natural Questions (Kwiatkowski et al., 2019), which contains samples derived from Google Search queries, covering a broad range of questions real-world users ask online; HotpotQA (Yang et al., 2018), a multi-hop QA dataset that requires chaining multiple passages to answer questions; and MS MARCO (Bajaj et al., 2018), a large-scale reading comprehension dataset derived from real Bing search queries. For *long-form QA*, we use WixQA (Cohen et al., 2025), an enterprise QA benchmark grounded in real customer support interactions from the Wix Help Center. For *fact verification*, we use SciFact-Open (Wadden et al., 2022), which targets open-domain scientific claim verification against a large corpus of research abstracts. We randomly sample 500 examples each from Natural Questions, HotpotQA, and MS MARCO. For WixQA and SciFact-Open, we use all available samples: 200 and 191, respectively, totaling 1,891 samples.

Evaluation Metrics. Previous work usually used string match metrics for answer evaluation (Mallen et al., 2023; Gao et al., 2023). However, such metrics are rigid and cannot reliably evaluate model performance. For *open-domain QA* tasks (Natural Questions, HotpotQA, MS MARCO), we combine traditional string matching with LLM judges. We first check whether the gold answer can be found as a substring in the model-generated output. For samples where the gold answer is not found, we use three LLM judges—GPT-5-mini (OpenAI, 2025), Gemini-2.5-Flash (Google, 2025), and Claude Haiku 4.5 (Anthropic, 2025a)—and take the majority vote to determine whether the generated answer semantically aligns with the gold answer (see the prompts in Appendix E). For *long-form QA* (WixQA), we follow the original authors and use ROUGE-2 recall (Lin, 2004) as the evaluation metric (Cohen et al., 2025). The *fact verification* task (SciFact-Open) only requires a binary judgment of whether the claim is supported by the retrieved documents, and thus we use the exact match of the model’s binary judgment with the gold label.

While using LLM judges for all tasks would be better (e.g., evaluating the coverage of reference facts for long-form QA), the large number of experimental setups (models $\times$ retrieval sizes $\times$ orders $\times$ retrievers $\times$ datasets) makes purely LLM-based evaluation prohibitively costly. As a supplementary analysis, in Appendix A.2, on a subset of WixQA, we report evaluation results that are based on LLM judges. In addition, we include an analysis of the agreement level between LLM judges in Appendix A.3.

Retrieval Corpus. For Natural Questions and HotpotQA, we use Wikipedia as the retrieval corpus. We processed the wikidump from November 2023, which contains 6 million articles. We split each article into chunks by double newlines, resulting in 20 million chunks. Each chunk is treated as an independent “document” for retrieval. For MS MARCO, WixQA, and SciFact-Open, we use the passage corpora provided with each respective dataset.

4.2 LLM Systems

Backbone LLMs. 11 LLMs from two open weight families and three proprietary families are tested, including Qwen3 (4B, 8B, 32B) (Team, 2025), Llama-3 Instruct (3.1-8B,

3.3-70B) (Meta, 2024a;b), OpenAI GPT-5 and GPT-5-mini (OpenAI, 2025), Google Gemini-2.5-Pro and Gemini-2.5-Flash (Google, 2025), and Anthropic Claude Sonnet 4.5 and Claude Haiku 4.5 (Anthropic, 2025a). After extending the context length of Qwen3 models to 128K with RoPE scaling following official recommendations, all models are able to consume the maximum retrieval size of 100 documents.

Prompting Strategies. Besides the vanilla prompting strategy that concatenates all retrieved documents in the user prompt, we explore four alternative strategies that vary how models interact with the retrieved documents:

OwnKnow obtains a draft answer based on models’ own knowledge by prompting without retrieval in the first step, and then inserts this draft answer into the prompt for the RAG setup. This resembles how Astute RAG (Wang et al., 2025) elicits the LLM’s internal knowledge.

Most of the latest LLMs have reasoning capability. **NoThink** disables or minimizes the model’s reasoning budget (i.e., sets the thinking token budget to the lowest possible), testing how disabling reasoning impacts models’ retrieval robustness profiles.

As using LLMs as tool-use agents has become one of the most prevalent usage patterns, we also examine how structuring retrieval results as tool responses affects the robustness profile. **ForcedTool** creates a dummy retrieval tool and forces the model to call it, and the retrieved documents are then provided as tool responses rather than directly in the user prompt. **AutoTool** is similar to ForcedTool but lets the model decide whether to call the retrieval tool, which can be considered as an agent variant of the OwnKnow strategy.

As Llama models do not natively support tool usage and adjustment of reasoning content, we only test Llama models with vanilla prompting and OwnKnow.

4.3 RAG Parameters

Retrievers. Our retrieval system is built on top of Apache Solr 9¹. We use two retrievers: one is the canonical sparse retriever based on BM25 (Robertson & Zaragoza, 2009), and the other is a cosine similarity based dense retriever where we embedded each document by bge-large-en-v1.5² (Xiao et al., 2023). For any robustness metric defined in Section 3, we get the results for both retrievers and take the average.

Sizes. We experiment with retrieval sizes of 5, 10, 25, 50, 75, and 100 documents. The retrieval size is capped at 100 documents as most models have reached their maximum context lengths. When the retrieved documents exceed the maximum context length of a model, we iteratively drop the lowest ranked document.

Orders. For each of these sizes, we apply three ordering strategies based on the retriever’s ranking of the documents: the **original** order (returned by the retriever), the **reversed** order (reversing the original order), and a **reranked** order produced by the BGE reranker (Chen et al., 2024a).³ We test the reversed

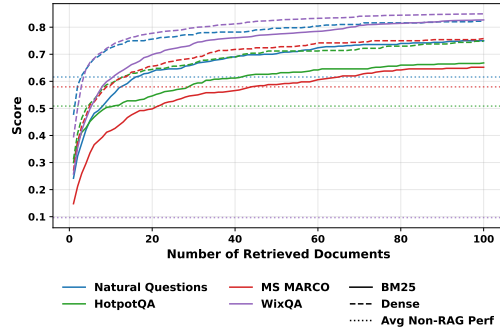


Figure 3: Performance of the retrievers, measured by the recall of gold answers within the concatenated retrieved documents. For open-domain QA, the gold answer is considered covered if any of its alternative forms exactly match a substring in the concatenated retrieved documents. For WixQA, we calculate the ROUGE-2 recall.

¹<https://solr.apache.org/docs/9.0.0/index.html>

²<http://huggingface.co/BAAI/bge-large-en-v1.5>

³<https://huggingface.co/BAAI/bge-reranker-v2-m3>

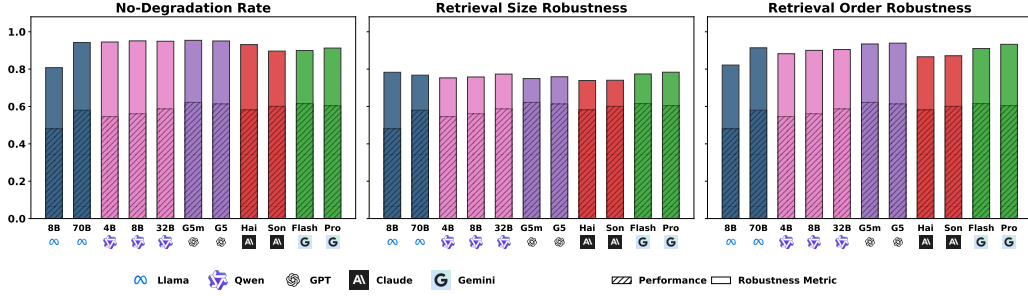


Figure 4: The three retrieval robustness metrics and task performance of experimented LLMs using vanilla prompting. G5m: GPT-5 Mini; G5: GPT-5; Hai: Haiku; Son: Sonnet. The mean of task performance achieved with different retrieval sizes and orders is shown for each model. Model families are indicated by icons, while the variants are indicated by model sizes or names. Models generally have good retrieval robustness.

order because sometimes we want to put the most relevant document at the end of the prompt (the closest to the answer). We include the reranked order to simulate a realistic reranking stage on top of the retriever.

Retrieval Quality. As our retrieval robustness benchmark relies on the retrievers, we examine the retrieval quality by checking the recall of gold answers within the retrieved documents. We follow prior work and determine that the concatenated retrieved documents contain the gold answer if its substring is an exact match of any form of the gold answer (substring exact match) (Mallen et al., 2023). For WixQA, we calculate the ROUGE-2 recall by taking the concatenated retrieved documents as the hypotheses. SciFact is excluded from this retrieval quality analysis as it only contains gold binary labels. For reference, we also report the average model performance without RAG (Non-RAG Perf). As shown in Figure 3, both retrievers provide sufficiently high-quality retrieval, ensuring that the findings of our experiments are based on valid setups.

5 Results

5.1 Overall Robustness

We report the three retrieval robustness metrics for LLM systems using vanilla prompting in Figure 4. Besides robustness, task performance is shown in the same figure with bars with a different hatch style. Retrieval robustness is calculated following the definitions in Section 3, while task performance is the average score across all retrieval size k , retrieval order o , retrievers, and datasets. **All models achieve higher than 75% retrieval robustness across all metrics.** Compared to prior studies that highlight the weak robustness of RAG systems under synthetic setups, such as using artificially created documents (Wu et al., 2024), we show that LLMs demonstrate surprisingly good retrieval robustness in more realistic settings.

Nevertheless, **retrieval robustness varies substantially across datasets** (Figure 5), suggesting that the decision to adopt RAG remains a case-by-case consideration. While models generally demonstrate high retrieval robustness on short-form question answering and fact verification, their robustness drops significantly on WixQA, which requires longer outputs. This is partly because models already show high variance when generating open-ended, long-form content, and the inclusion of many retrieved documents can further increase this variance. Furthermore, even on tasks where models are generally robust, the top-performing model is not consistent. For example, on short-form datasets, GPT-5 typically shows the best robustness. However, on MS MARCO, its smaller and more cost-effective variant actually surpasses it.

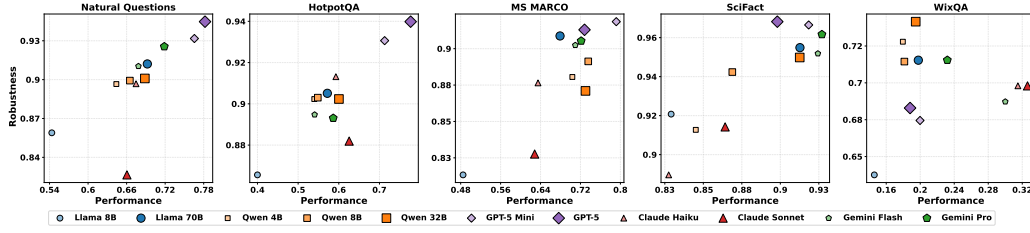


Figure 5: Geometric mean of the three retrieval robustness metrics compared against task performance for each model using vanilla prompting across various datasets. On WixQA, models exhibit low retrieval robustness because the task requires generating open-ended long-form outputs. These outputs are inherently high-variance and are further affected by the long input context constructed with the retrieved documents.

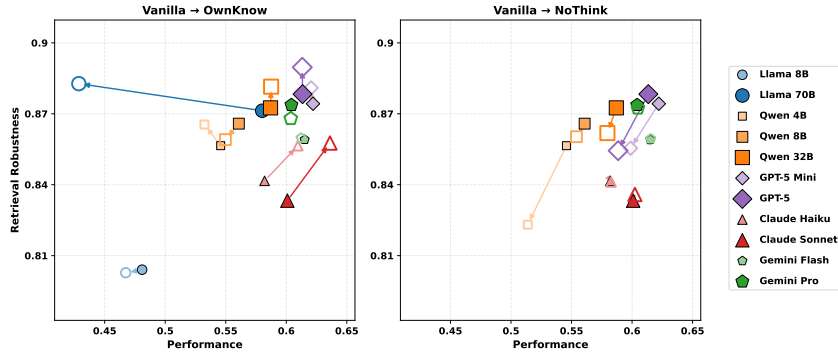


Figure 6: Effects of OwnKnow and NoThink settings. Shaded markers represent vanilla prompting, while transparent markers represent OwnKnow / NoThink. Effects are indicated by arrows pointing from the vanilla setting to OwnKnow / NoThink setting. Generally speaking, providing models with their prior knowledge and enabling reasoning improves robustness.

5.2 Effects of Prompting

Figure 6 visualizes how adding models’ prior knowledge to the context (**OwnKnow**) or disabling reasoning (**NoThink**) impacts model performance and retrieval robustness. OwnKnow generally provides modest robustness gains, as the non-RAG draft serves as an anchor that reduces variance in model outputs. This is analogous to self-refinement, which has been shown to be an effective prompting technique (Yang et al., 2022; Madaan et al., 2023). However, the inclusion of models’ prior knowledge naturally reduces the tendency of models to use information in the retrieved contexts, which can lead to drops of performance compared to the vanilla prompting. Besides the huge performance drop of Llama-3.3-70B as shown in the figure, we look into the maximum task performance across retrieval sizes rather than the mean task performance, and observe that all models except for the Claude family yield a lower maximum task performance when using OwnKnow.

Among models explicitly trained or fine-tuned for reasoning, Qwen and GPT experience more pronounced declines in both robustness and performance than other model families when reasoning mode is disabled. A detailed breakdown across various datasets (Figure 17 in Appendix B.2) reveals that while both Qwen and GPT experience the expected substantial performance and robustness drops on HotpotQA, a multi-hop QA dataset that requires some reasoning, they also suffer substantial robustness declines on simpler tasks. Most concerning is their robustness on single-hop datasets, including Natural Questions and MS MARCO, which suggests these models may have become overly dependent on their reasoning context even for basic RAG tasks.

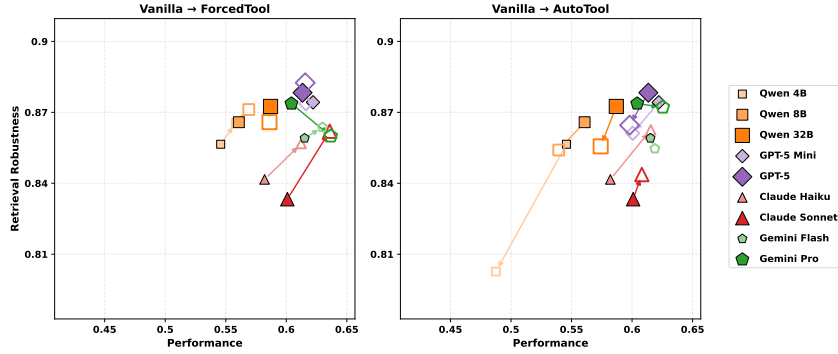


Figure 7: Effects of tool usage. Models are required to call the given retrieval tool in ForcedTool setting and can decide whether to call the retrieval tool in AutoTool setting. Retrieved documents are then provided as tool responses.

5.3 Effects of Tool Usage

Claude models consistently demonstrate better robustness and performance when retrieved documents are treated as tool responses, as shown in Figure 7. This suggests that Claude models are highly optimized for tool-based interactions, which is further evidenced by their strong performance in agentic applications like Claude Code (Anthropic, 2025b). Consequently, we hypothesize that injecting additional context through tool calls and responses, even through dummy tools, is likely a more effective integration strategy for Claude.

In contrast, Qwen and GPT models exhibit noticeable drops in robustness and performance when tool calls are not strictly required. Compared to Claude and Gemini models, Qwen and GPT models utilize the retrieval tool significantly less frequently. For instance, in the same setup, while Claude and Gemini models trigger retrieval over 99% of the time, Qwen-4B and GPT-5 do so only 87% and 33% of the time, respectively. Notably, this lower frequency of tool invocation does not inherently account for the drop of robustness. When the retrieval tool is bypassed, these models rely on their internal knowledge and should, in theory, produce outputs similar to their non-RAG baselines. We instead posit that the reduced robustness of Qwen and GPT under the AutoTool setting stems from a lack of robustness to the additional tool schema in the context, leading to varying outputs.

5.4 Oracle Performance

Lastly, we investigate the oracle performance when models have perfect retrieval robustness across k s. Concretely, for each sample, we keep the best output the model generates so far until the current k across all retrieval orders.

As illustrated by Figure 8, having perfect retrieval robustness can lead to 8% and 15% of absolute performance improvement for GPT-5 and Qwen-4B. Notably, Qwen-4B can even reach similar performance to GPT-5, suggesting the importance of retrieval robustness.

5.5 Performance-Weighted Robustness

While retrieval robustness is orthogonal to task performance and provides complemen-

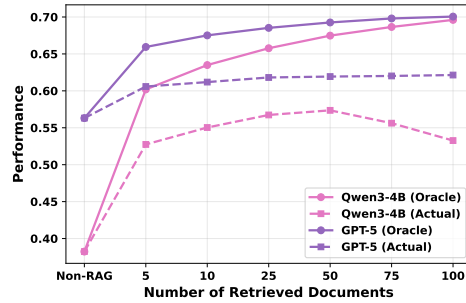


Figure 8: Average task performance of models using vanilla prompting with actual retrieval robustness (Actual) and perfect retrieval robustness (Oracle) across k s. Perfect retrieval robustness means the model would not produce outputs that are worse than it has previously produced with smaller values of k .

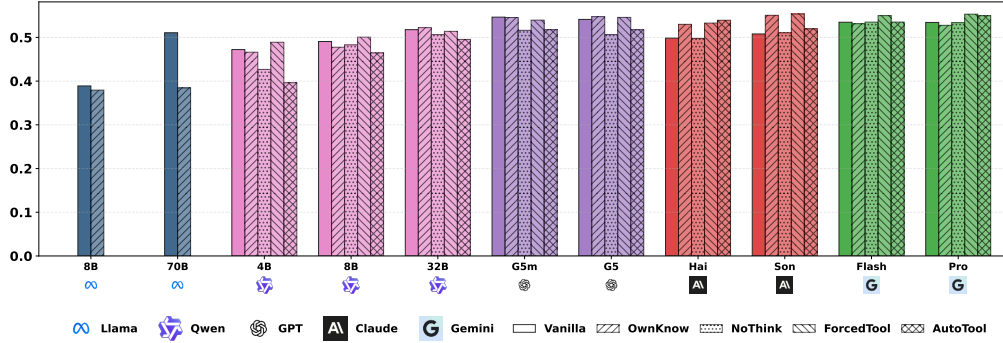


Figure 9: Performance-Weighted Robustness (PWR) across all five prompting setups. PWR combines robustness and task performance into a single score, penalizing models that achieve high robustness simply by being consistently wrong. Llama models are only tested with vanilla and OwnKnow setups.

tary dimensions for model evaluation, a model that consistently gives incorrect answers can achieve perfect robustness. To mitigate confusion for model selection, we consider **Performance-Weighted Robustness (PWR)**, which jointly accounts for both retrieval robustness and task performance:

$$\text{PWR} = (\text{NDR} \cdot \text{RSR} \cdot \text{ROR})^{1/3} \times \text{Performance} \quad (4)$$

where Performance is the average task score across all retrieval sizes, orders, retrievers, and datasets. Multiplying retrieval robustness by performance ensures that only models that are both robust and accurate receive high scores.

Figure 9 shows PWR across all five prompting setups. Llama-3.3-70B, under the OwnKnow setup, is significantly penalized for its low task performance despite maintaining good robustness compared to the vanilla setup (Figure 6). While GPT-5 has almost the same task performance under vanilla and OwnKnow setups, the better retrieval robustness leads to higher PWR. Among the setups, **ForcedTool yields the highest PWR** for most models, further highlighting the benefit of formulating retrieval as tool calls and tool responses.

6 Conclusions

We introduce retrieval robustness metrics—no-degradation rate, retrieval size robustness, and retrieval order robustness—to quantify how reliably LLMs handle queries via RAG. A realistic benchmark of 1,891 samples is compiled, spanning five datasets across three task categories, with documents retrieved using both sparse and dense retrievers. Our experiments with 11 LLMs from 5 families reveal that models generally achieve high retrieval robustness, but robustness varies substantially across tasks, with long-form generation being notably less robust than short-form QA and fact verification. Through four additional prompting strategies, we find that reasoning-trained models such as Qwen and GPT suffer notable robustness declines when reasoning is disabled, even on simple tasks, and that tool-based retrieval integration benefits Claude models but hurts Qwen and GPT. We hope our benchmark and findings inspire further research on robust RAG systems.

Reproducibility Statement

Questions in our benchmark come from Natural Questions (Kwiatkowski et al., 2019) (Hugging Face⁴), HotpotQA (Yang et al., 2018) (Hugging Face⁵), MS MARCO (Bajaj et al., 2018)

⁴<https://huggingface.co/datasets/google-research-datasets/natural-questions>

⁵<https://huggingface.co/datasets/hotpotqa/hotpot-qa>

(Hugging Face⁶), SciFact-Open (Wadden et al., 2022) (Hugging Face⁷), and WixQA (Cohen et al., 2025) (Hugging Face⁸). We will release code to reproduce our benchmark, including the script for obtaining sampled questions and retrieval corpus. We will also release the script for calculating retrieval robustness metrics given model outputs.

Most of our experiment settings can be found in Section 4, with inference details available in Appendix C. Furthermore, we include the prompt templates we use in our experiment in Appendix E.

References

- Anthropic. Claude 4.5 sonnet and claude 4.5 haiku, 2025a. URL <https://www.anthropic.com/news/claude-4-5-sonnet>.
- Anthropic. Claude code, February 2025b. URL <https://www.anthropic.com/claude-code>.
- Anthropic. claude-3-5-sonnet, July. 2024. URL <https://www.anthropic.com/news/claude-3-5-sonnet>.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*, 2024.
- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension dataset, 2018. URL <https://arxiv.org/abs/1611.09268>.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1171. URL <https://aclanthology.org/P17-1171>.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024a.
- Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17754–17762, 2024b.
- Dvir Cohen, Lin Burg, Sviatoslav Pykhnivskyi, Hagit Gur, Stanislav Kovynov, Olga Atzmon, and Gilad Barkan. Wixqa: A multi-dataset benchmark for enterprise retrieval-augmented generation, 2025. URL <https://arxiv.org/abs/2505.08643>.
- Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for RAG systems. *arXiv preprint arXiv:2401.14887*, 2024.
- Feiteng Fang, Yuelin Bai, Shiwen Ni, Min Yang, Xiaojun Chen, and Ruifeng Xu. Enhancing noise robustness of retrieval-augmented language models with adaptive adversarial training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.
- Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp.

⁶https://huggingface.co/datasets/microsoft/ms_marco

⁷<https://huggingface.co/datasets/davidheineman/scifact-open>

⁸<https://huggingface.co/datasets/Wix/WixQA>

- 6465–6488, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.398. URL <https://aclanthology.org/2023.emnlp-main.398/>.
- Google. Gemini 2.5, 2025. URL <https://deepmind.google/technologies/gemini/>.
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong 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 (NAACL)*, 2024.
- Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL <https://aclanthology.org/D17-1215>.
- Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl.a.00276. URL <https://aclanthology.org/Q19-1026/>.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS ’20*, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.
- Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL <https://aclanthology.org/W04-1013/>.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023. URL <https://arxiv.org/abs/2303.17651>.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL <https://aclanthology.org/2023.acl-long.546/>.
- Meta. Introducing llama 3.1, Jul 2024a. URL <https://ai.meta.com/blog/meta-llama-3-1/>.
- Meta. Llama 3.3, 2024b. URL <https://ai.meta.com/blog/llama-3-3/>.
- OpenAI. Gpt-5, 2025. URL <https://openai.com/index/gpt-5/>.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Gertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit

- Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermeni, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghamman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. Gpt-4o system card, 2024. URL <https://arxiv.org/abs/2410.21276>.
- Ruotong Pan, Boxi Cao, Hongyu Lin, Xianpei Han, Jia Zheng, Sirui Wang, Xunliang Cai, and Le Sun. Not all contexts are equal: Teaching LLMs credibility-aware generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.
- Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009. ISSN 1554-0669. doi: 10.1561/15000000019. URL <https://doi.org/10.1561/15000000019>.
- Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org, 2023.
- Qwen Team. Qwen3 technical report, 2025. URL <https://qwenlm.github.io/blog/qwen3/>.
- Nandan Thakur, Luiz Bonifacio, Crystina Zhang, Odunayo Ogundepo, Ehsan Kamaloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Boxing Chen, Mehdi Rezagholizadeh, and Jimmy Lin. “knowing when you don’t know”: A multilingual relevance assessment dataset for robust retrieval-augmented generation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 12508–12526, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.730. URL <https://aclanthology.org/2024.findings-emnlp.730>.
- David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. SciFact-open: Towards open-domain scientific claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 4719–4734, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.347. URL <https://aclanthology.org/2022.findings-emnlp.347>.
- Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Serkan Ö. Arik. Astute RAG: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.
- Zhepei Wei, Wei-Lin Chen, and Yu Meng. InstructRAG: Instructing retrieval-augmented generation via self-synthesized rationales. *arXiv preprint arXiv:2406.13629*, 2024.
- Jason Weston and Sainbayar Sukhbaatar. System 2 attention (is something you might need too), 2023. URL <https://arxiv.org/abs/2311.11829>.
- Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. How easily do irrelevant inputs skew the responses of large language models? In *First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=S7NVVfuRv8>.

- Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.
- Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. In *International Conference on Learning Representations (ICLR)*, 2024.
- Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. Re3: Generating longer stories with recursive reprompting and revision. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 4393–4479, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.296. URL <https://aclanthology.org/2022.emnlp-main.296/>.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun’ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL <https://aclanthology.org/D18-1259/>.
- Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=ZS4m74kZpH>.
- Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.
- Linda Zeng, Rithwik Gupta, Divij Motwani, Yi Zhang, and Diji Yang. Worse than zero-shot? a fact-checking dataset for evaluating the robustness of RAG against misleading retrievals. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL <https://openreview.net/forum?id=R4MeWTeVej>.

A Additional Analysis

A.1 Soft Trend-based Retrieval Size Robustness

We consider a softer, trend-based alternative to retrieval size robustness, which measures the overall monotonic trend of the performance trajectory as retrieval size increases. Formally, for a fixed example q , document order o , and the set of retrieval sizes $K = \{k_1, \dots, k_n\}$ with $k_1 < \dots < k_n$, we define:

$$\text{Trend-RSR}_{(q,o)} = \frac{\tau_{(q,o)} + 1}{2} \quad (5)$$

where $\tau_{(q,o)}$ is Kendall’s rank correlation coefficient between the retrieval-size index and the corresponding performance scores:

$$\tau_{(q,o)} = \frac{1}{\binom{n}{2}} \sum_{i < j} \text{sgn}(f(q, k_j, o) - f(q, k_i, o)) \quad (6)$$

The $(\tau + 1)/2$ rescaling maps the correlation from $[-1, 1]$ to $[0, 1]$: a score of 1.0 indicates a perfectly non-decreasing trajectory (performance never drops as k increases), 0.5 indicates no systematic trend, and values below 0.5 indicate a predominantly decreasing trend.

Results. Figure 10 compares soft RSR against the original strict RSR for the vanilla prompting setup. Soft RSR is consistently higher than strict RSR, as expected since it tolerates local dips that the strict formulation penalizes. Notably, the *model ranking is preserved*: models that score high (or low) on strict RSR also score high (or low) on Soft RSR.

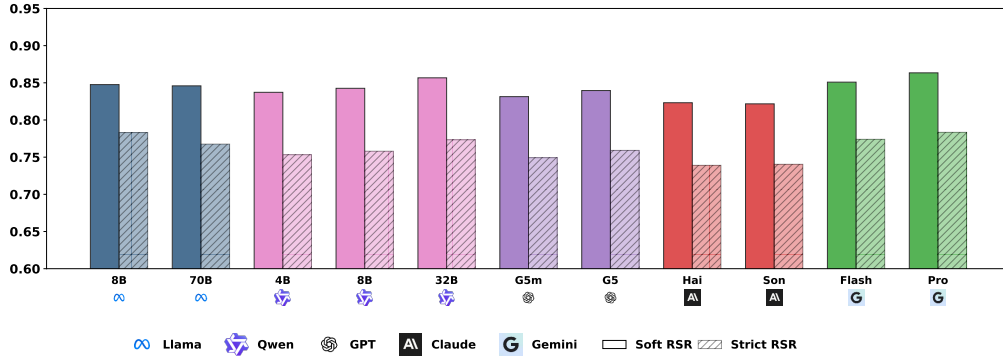


Figure 10: Soft RSR vs. Strict RSR per model. Models have higher RSR under the softer formulation, while their ranks are consistent across the two formulations.

A.2 WixQA with LLM Judge Evaluation

We use ROUGE-2 recall instead of LLM judges for WixQA in the main paper due to the high cost of running LLM judges on the full dataset. Here we conduct a supplementary analysis using LLM-based fact recall for evaluation on a subset of WixQA.

Setup. We decompose each reference answer into atomic facts using Claude Sonnet 4.5, then use three models (GPT-5-mini, Claude Haiku 4.5, and Gemini 2.5 Flash) to determine whether each reference fact is covered by the model’s prediction. The per-example fact recall score is the fraction of reference facts judged as covered, averaged across the three judges. We evaluate the first 50 examples across all 11 models, all 6 retrieval sizes, all 3 document orders, using the BM25 retriever under the vanilla prompting setup.

Robustness Comparison. Figure 11 compares the three robustness metrics computed using ROUGE-2 recall versus LLM-judge fact recall on the subset. The relative model rankings are moderately preserved across both metrics (Spearman $\rho = 0.79, 0.59$, and 0.45 for the three robustness metrics).

Overall, long-form QA remains the task where models face the most substantial retrieval robustness challenges, with RSR consistently around 0.47 regardless of the scoring metric.

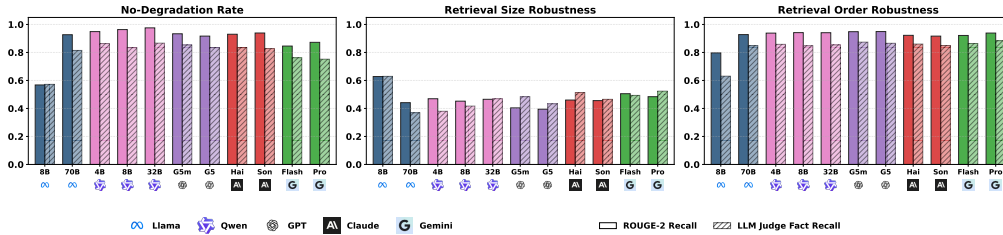


Figure 11: Robustness metrics on WixQA (50-example subset, BM25, vanilla setup) computed using ROUGE-2 recall (solid) vs. LLM-judge fact recall averaged over 3 judges (hatched). Low RSR on long-form QA is observed across both task evaluation metrics.

A.3 Agreement Level among LLM Judges

For open-domain QA evaluation, we use three LLM judges from different model families (GPT-5-mini, Claude Haiku 4.5, and Gemini 2.5 Flash) to determine answer correctness via majority vote (§4.1). We investigate if the three judges produce consistent judgment, as a proxy measure of LLM judge quality. Across all models, datasets, retrieval sizes, document orders, and retrievers in our benchmark, the judges collectively annotate 610,500 samples.

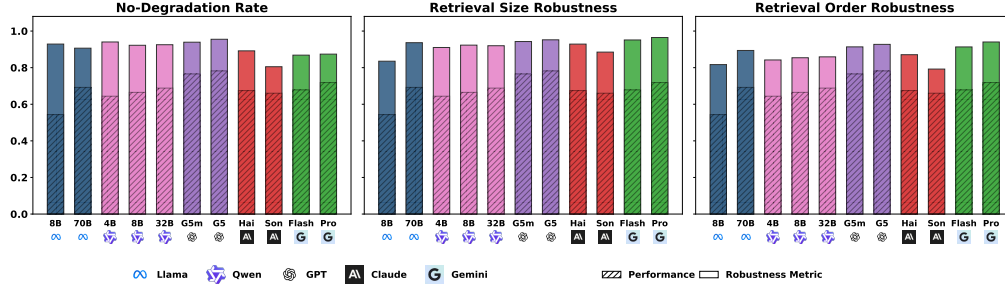


Figure 12: The three retrieval robustness metrics and task performance of experimented LLMs using vanilla prompting on Natural Questions.

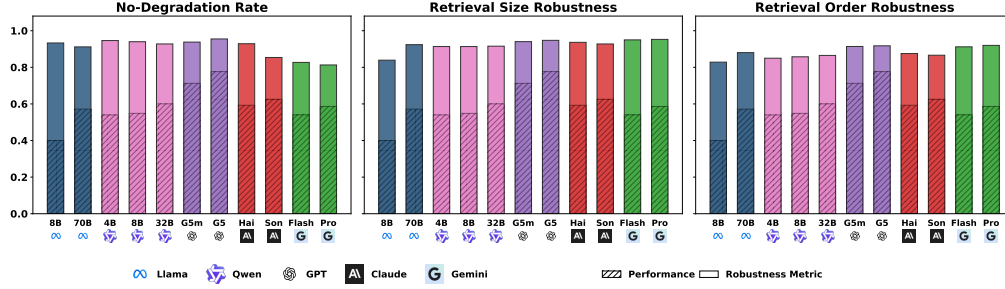


Figure 13: The three retrieval robustness metrics and task performance of experimented LLMs using vanilla prompting on HotpotQA.

As shown in Table 1, the three judges achieve high inter-annotator agreement: Fleiss’ $\kappa = 0.83$ and an average pairwise agreement rate of 91.8%.

Table 1: Inter-judge agreement statistics on open-domain QA evaluation.

Judge 1	Judge 2	Agreement	Cohen’s κ
GPT-5-mini	Claude Haiku 4.5	0.906	0.807
GPT-5-mini	Gemini 2.5 Flash	0.946	0.887
Claude Haiku 4.5	Gemini 2.5 Flash	0.901	0.797
Overall (all 3 judges)			
Fleiss’ κ		0.830	
Avg pairwise agreement		0.918	

B Dataset Breakdown Results

B.1 Retrieval Robustness

We show the retrieval robustness metrics and average RAG performance in Figures 12, 13, 14, 15, and 16.

B.2 Prompting Strategy Effects

We show the effects of NoThink, OwnKnow, ForcedTool, and AutoTool in Figures 17, 18, 19, and 20.

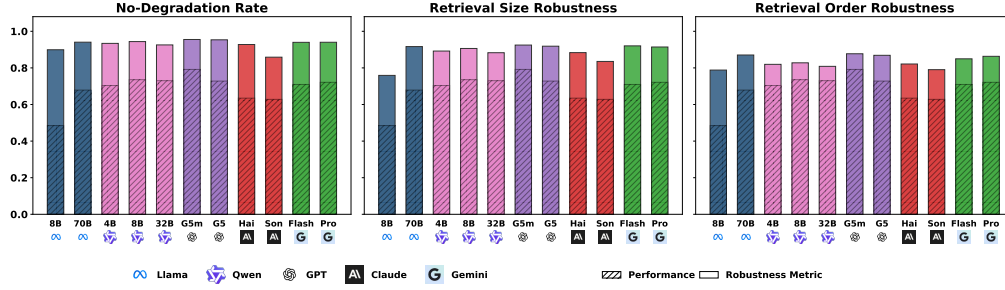


Figure 14: The three retrieval robustness metrics and task performance of experimented LLMs using vanilla prompting on MS MARCO.

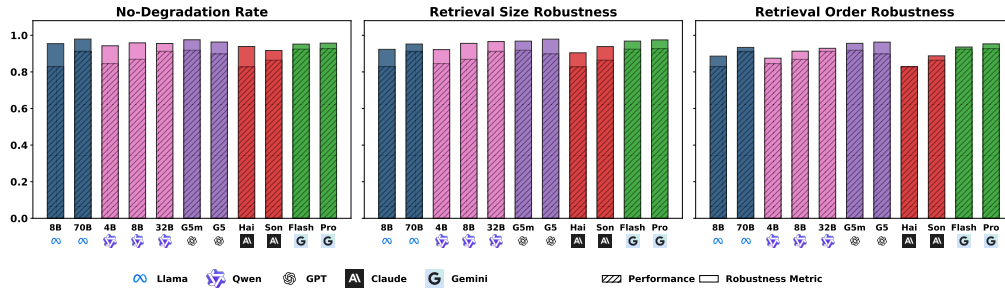


Figure 15: The three retrieval robustness metrics and task performance of experimented LLMs using vanilla prompting on SciFact-Open.

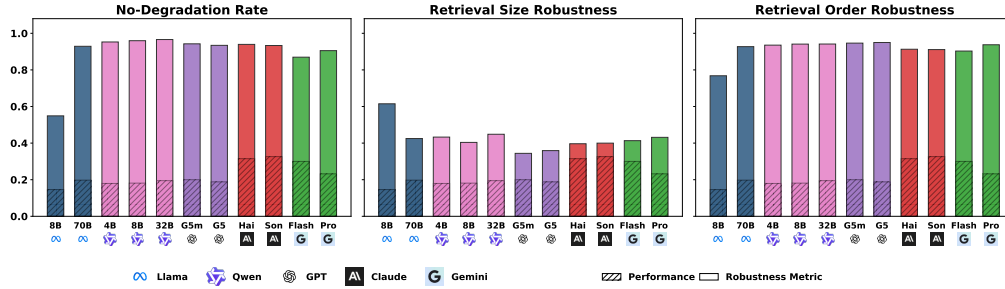


Figure 16: The three retrieval robustness metrics and task performance of experimented LLMs using vanilla prompting on WixQA.

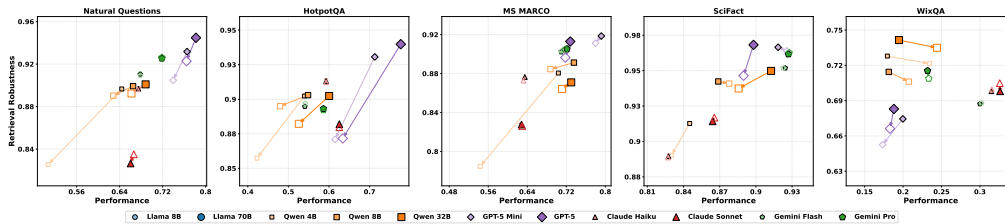


Figure 17: Effects of NoThink on different datasets.

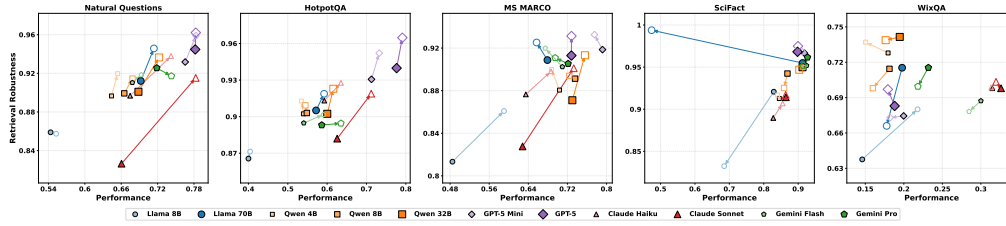


Figure 18: Effects of **OwnKnow** on different datasets.

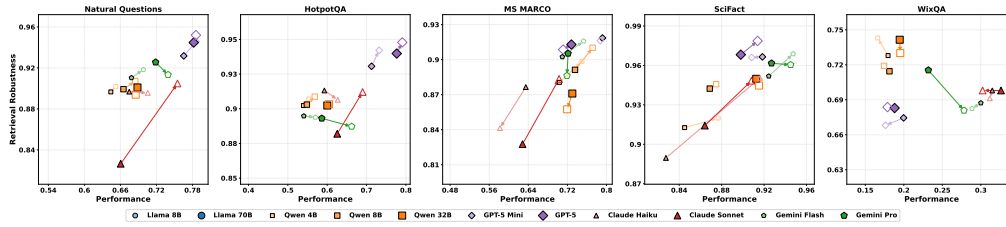


Figure 19: Effects of **ForcedTool** on different datasets.

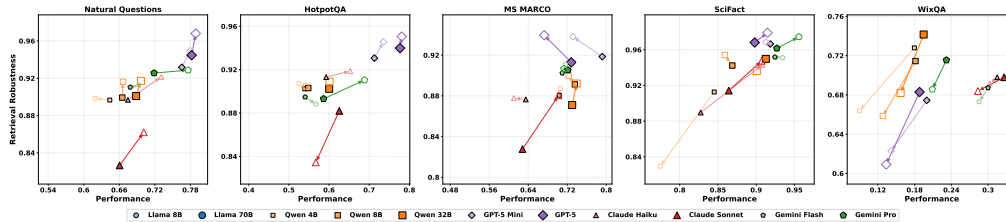


Figure 20: Effects of **AutoTool** on different datasets.

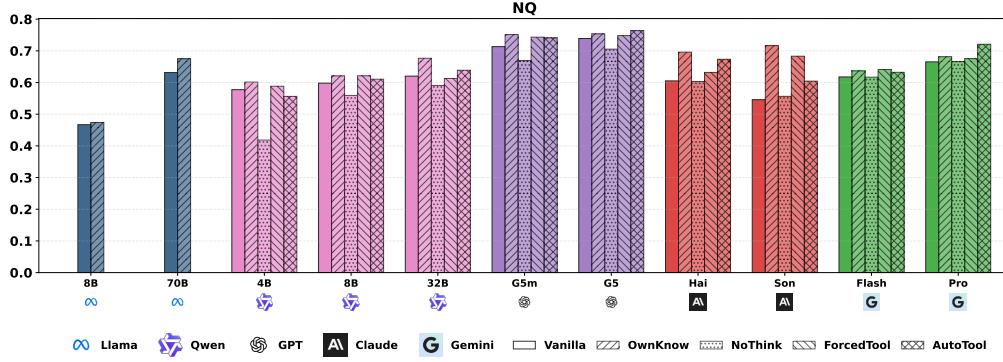


Figure 21: Performance-weighted robustness on Natural Questions.

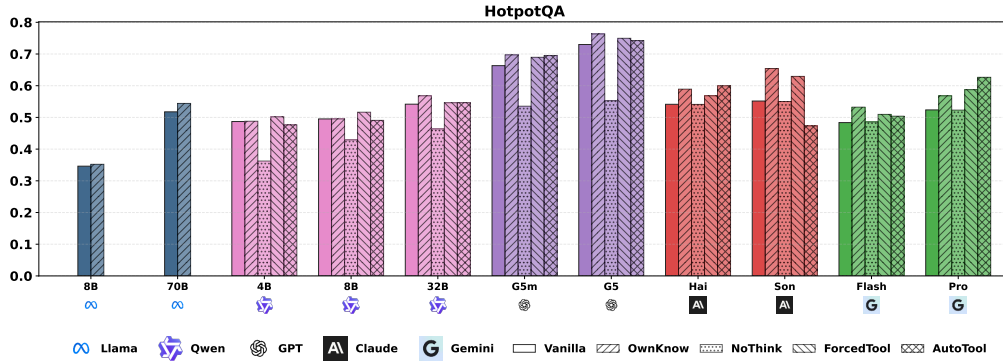


Figure 22: Performance-weighted robustness on HotpotQA.

B.3 Performance-Weighted Robustness

We show the performance-weighted robustness on the 5 datasets in Figures 21, 22, 23, 24, and 25.

C Inference Setup

Inference Parameters. Due to the computational cost and running time, we use greedy decoding and perform inference with each model under each setup once. However, we note that for proprietary models, setting temperature to 0 would not result in deterministic

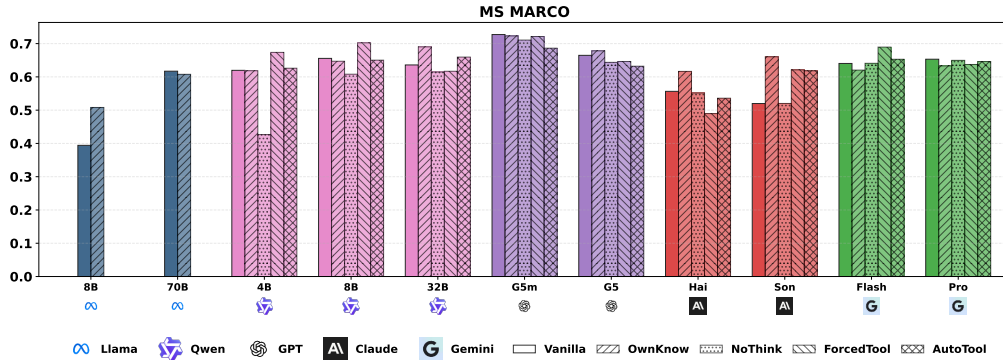


Figure 23: Performance-weighted robustness on MS MARCO.

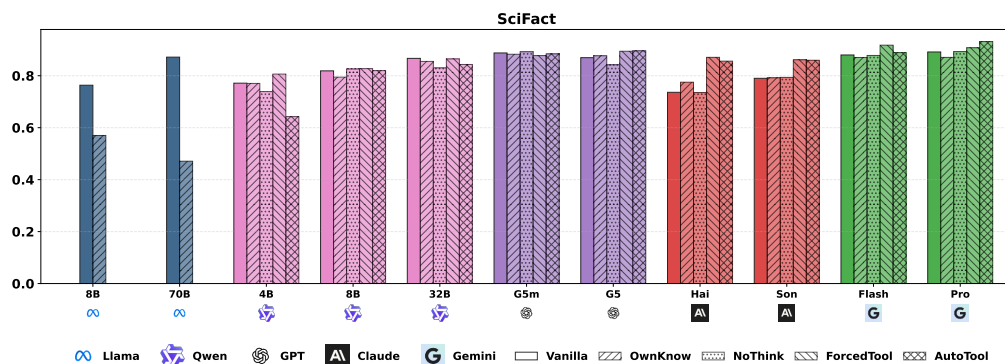


Figure 24: Performance-weighted robustness on SciFact-Open.

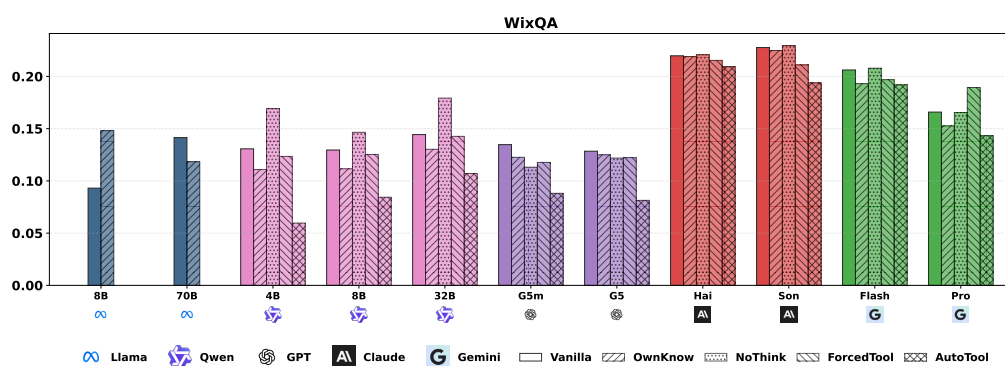


Figure 25: Performance-weighted robustness on WixQA.

decoding. During inference, models are allowed to generate at most 4096 tokens on all datasets, except for WixQA where we allow models to generate 8192 tokens.

Inference Infrastructure. For models with publicly available weights, we use vLLM for more efficient inference (Kwon et al., 2023), and our experiments are conducted on compute nodes with 8 H100 GPUs.

D The Use of LLMs

We use LLMs (Gemini, Claude) to polish writing, including correcting grammar errors, making language more concise, and plotting figures.

E Prompt Templates

The prompt templates (in jinja2 format) used in our experiments can be found at the end of the Appendix.

non_rag.qa.j2

```
1 Answer the user given question in a concise manner. Indicate the answer
  with Answer: and only include the answer words or phrases.<eosys>{{
    question }}
```

rag.qa.j2

```
1 Answer the user given question in a concise manner. Indicate the answer
  with Answer: and only include the answer words or phrases. The user
  would also provide retrieved documents that might be useful for
  answering the question.<eosys>Retrieved Documents:
2 {% for source in sources %}Document {{ loop.index }}:
3 {{ source }}
4
5 {% endfor %}}{{ question }}
```

rag.qa.ownknow.j2

```
1 Answer the user given question in a concise manner. Indicate the answer
  with Answer: and only include the answer words or phrases. The user
  would also provide retrieved documents that might be useful for
  answering the question. In addition, the user would also provide your
  previously made answer (if available) which you can revise.<eosys>
  Retrieved Documents:
2 {% for source in sources %}Document {{ loop.index }}:
3 {{ source }}
4
5 {% endfor %}}{{ question }}
6 Previous Answer: {{ open_qa_output }}
```

rag.qa.tool.j2

```
1
2 {{ SYSTEM_START }}
3 Answer the user given question in a concise manner. Indicate the answer
  with Answer: and only include the answer words or phrases. Access
  to a retriever is provided to gather relevant documents. You are
  allowed to call the retriever at most once.
4 {{ SYSTEM_END }}
5 {{ USER_START }}
6 {{ question }}
7 {{ USER_END }}
```

answer_match.j2

```
1 {{ SYSTEM_START }}
2 You will be given a question, a list of gold answers to this question,
  and a predicted answer. Any answer from the gold answer list is
  considered a correct answer to the question. Your task is to judge
  whether the predicted answer can match any of the gold answers.
3 Note that predicted answer does not have to exactly match the gold
  answers. It is considered a match as long as the predicted answer is
  semantically equivalent to any of the gold answers.
4
5 Your response should be in the following JSON format:
6 {
7   rationale : Your rationale for the judgment as to whether prediction
  is matching any of the gold answers. ,
8   matching : 1 if the prediction matches any of the gold answers, 0
  otherwise.
9 }
10 {{ SYSTEM_END }}
11 {{ USER_START }}
```

```
12 QUESTION:
13 {{rendered_query}}
14
15 GOLD ANSWER:
16 {{gold_answer}}
17
18 PREDICTION:
19 {{prediction}}
20 {{ USER_END }}
```