Enhancing RAG Robustness: A Systematic Framework for Failure Diagnosis and Mitigation

Anonymous ACL submission

Abstract

The practical reliability of Retrieval-Augmented Generation (RAG) is often undermined by a spectrum of subtle and unpredictable failures, forcing developers into a costly trial-and-error process due to the absence of a systematic diagnostic framework. To address this critical gap, we introduce a comprehensive framework to systematically enhance RAG system robustness. Our methodology is twofold: first, we propose a detailed RAG Failure Taxonomy that categorizes five key failure modes—Negative Refusal, Ranking Confusion, Answer Absence, Noise Impact, and Complex Reasoning. Second, we present a corresponding suite of targeted Optimization Strategies designed to mitigate each specific failure. Through extensive experiments, we empirically validate our framework, demonstrating, for instance, that a two-step reasoning process reduces incorrect refusals on valid contexts from 60% to 10%, and that targeted query rewriting substantially outperforms more complex reasoning strategies. Our work provides a principled methodology and a practical toolkit for developers to diagnose, understand, and systematically improve the reliability of RAG systems.

1 Introduction

LLMs have demonstrated remarkable capabilities across a vast array of natural language tasks, yet they are fundamentally constrained by their static, pre-trained knowledge and a propensity for factual inaccuracies, commonly known as "hallucinations"(Brown et al., 2020; Ji et al., 2023; Achiam et al., 2023; Yang et al., 2025). RAG has emerged as a powerful and widely adopted paradigm to address these shortcomings(Zhao et al., 2023; Es et al., 2024). By grounding LLM responses in external, verifiable information sources, RAG promises to deliver more reliable, up-to-date, and trustworthy answers, making it a cornerstone for building next-generation AI systems(Guu et al., 2020; Shen et al., 2023).

However, the theoretical promise of RAG is often undermined in practice by a spectrum of reliability issues(Coman et al., 2025; Bai et al., 2023; **?**). These systems can fail in subtle and unpredictable ways that degrade performance and erode user trust(Bender et al., 2021; Müller et al., 2025). For instance, they may fail to surface the most relevant information to the top of the context (*Ranking Confusion*), overlook critical details even when they are provided (*Answer Absence*), or generate plausible-sounding but incorrect answers when faced with queries beyond the scope of the available context (*Negative Refusal*). Such failures pose a significant barrier to the deployment of RAG systems in real-world high-stakes applications(Sivasothy et al., 2025; Zhang et al., 2025b).

Although numerous studies have proposed point solutions to improve specific aspects of RAG(**?**Wang et al., 2023), such as developing more sophisticated retrieval algorithms or re-ranking models(Jiang et al., 2023; Arivazhagan et al., 2023), the field lacks a holistic framework for systematically diagnosing the root causes of these failures and applying targeted interventions. This piecemeal approach leaves developers and researchers with a trial-and-error process, without a principled methodology to debug and enhance their systems robustly.

To bridge this critical gap, we introduce a comprehensive framework for the systematic diagnosis and mitigation of common RAG failures. Our work makes the following key contributions:

- We propose a comprehensive **RAG Fail-**

ure Taxonomy**, which systematically categorizes five critical failure modes: Negative Refusal, Ranking Confusion, Answer Absence, Noise Impact, and Complex Reasoning.

- We design and curate a suite of **Targeted Diagnostic Datasets**, each paired with specific evaluation metrics, to reliably trigger and quantify these individual failure modes in a controlled environment.
- We present and empirically evaluate a corresponding suite of **Optimization Strategies**, providing a practical and verifiable toolkit for mitigating each identified failure.

The remainder of this paper is organized as follows. Section 3 details our proposed taxonomy and optimization strategies. Section 4 presents our extensive empirical evaluation. Finally, we conclude the paper and discuss future work.

2 Related Work

RAG has become a standard method for grounding LLMs in external knowledge to improve factuality and timeliness (Lewis et al., 2020; Gao et al., 2023; Singh et al., 2025). Research in this domain has largely advanced along two parallel tracks: enhancing RAG architectures and evaluating their performance(Press et al., 2022; Asai et al., 2024; Guan et al., 2025; Maynez et al., 2020; Chen et al., 2024). Our work integrates these two tracks by proposing a systematic framework to diagnose and mitigate common RAG failures.

The evolution from "naive" to "Advanced RAG" has introduced a modular pipeline with optimizations at various stages. Pre-retrieval techniques like query rewriting (Ma et al., 2023; Lin et al., 2023) and decomposition (Press et al., 2022; Singh et al., 2023) aim to refine user intent. Core retrieval has advanced with hybrid search methods combining sparse and dense retrievers (Karpukhin et al., 2020; Liu et al., 2024). Post-retrieval strategies, most notably re-ranking, are now widely used to combat the "lost-in-the-middle" problem (Liu et al., 2023), where models struggle with information buried in long contexts. While these methods improve performance, they also make failure diagnosis a non-trivial task.

Concurrently, studies have begun to analyze specific RAG failure modes beyond end-to-end metrics(Es et al., 2024; Yang et al., 2024). These include issues of faithfulness and attribution(Li and Ng, 2024), robustness to noisy context(**?**), and challenges in multi-hop reasoning(Zhang et al., 2025a). However, these analyses often remain isolated, and the proposed optimization techniques are not systematically mapped to specific failure scenarios. This leaves practitioners with a collection of powerful but disconnected tools, forcing them into a costly trial-and-error process.

Crucially, our framework pairs this diagnostic taxonomy with a corresponding suite of optimization strategies, empirically demonstrating a clear link between each failure and its targeted mitigation. Our contribution is a practical, structured methodology that aims to move RAG debugging from an ad-hoc, trial-and-error process to a more systematic and principled approach.

3 Methodology

To systematically address the reliability challenges inherent in RAG systems, we propose a two-part methodology. First, to establish a structured framework for diagnosis, we develop a detailed **RAG Failure Taxonomy** (Section 3.1) that identifies and defines five key failure modes, as illustrated in Figure 1. To complement this taxonomy, we also curated a suite of **specialized diagnostic datasets** and evaluation methods designed to reliably trigger and quantify each failure mode (detailed in Section 4). Second, building upon this diagnostic framework, we introduce a corresponding suite of **Optimization Strategies** (Section 3.2), with each strategy designed to target and mitigate a specific failure. Together, these components form a comprehensive methodology for understanding, diagnosing, and enhancing the robustness of RAG systems.

3.1 RAG Failure Taxonomy

To build a foundation for systematic diagnosis, we establish a taxonomy of five key failure modes. The rationale for this classification is principled: each category corresponds to a failure at a distinct functional stage of the RAG workflow. This stage-based decomposition allows us to trace errors from initial query under-

standing and retrieval to the final answer synthesis. This ensures comprehensive diagnostic coverage. We define these modes as: Negative Refusal, Ranking Confusion, Answer Absence, Noise Impact, and Complex Reasoning.

Golden Case. In the context of RAG, a golden case is defined as a scenario where retrieval is both entirely accurate and complete. When the retrieved information is subsequently fed into the large model, it facilitates the generation of a correct response.

Negative Refusal. The challenge of negative refusal in RAG systems, where models tend to produce deceptive responses instead of acknowledging uncertainty, will severely erode user trust. The issue of negative refusal often stems from the model's lack of awareness of its knowledge boundaries. When confronted with a query that lacks sufficient information, the model may generate a factually incorrect or misleading response rather than transparently admitting the absence of relevant knowledge.

Ranking Confusion. Existing studies have shown that LLMs are more attentive to the earlier parts of input sequences. This characteristic poses a significant issue for RAG systems: even if the retrieval module successfully locates the correct document segments, the system may still be affected if the most relevant segments do not appear early in the input sequence. In such cases, the generation module will initially encounter suboptimal or erroneous information, ultimately impacting the accuracy and quality of the RAG system's output.

Answer Absence. The RAG system is designed to integrate information retrieved by the retrieval module with the inference capabilities of the generative model. However, a common challenge is that LLMs may overlook relevant details during the answer generation phase, even when all related contexts have been correctly retrieved. This issue often arises from the limitations in the reasoning capabilities of these models and problems associated with the format or order in which the context is presented.

Noise Impact. The robustness of RAG systems against noise is another critical aspect of their performance, especially when dealing with sources containing irrelevant or misleading information. Due to variations in the retrieval model's accuracy, the way users formulate their queries, and stylistic differences, the document chunks retrieved often contain some degree of irrelevant content, which, if fed into the LLMs, can significantly affect their reasoning performance and the final response.

Complex Reasoning. In practical tasks, handling complex reasoning scenarios that require integrating information from multiple documents is often necessary. In these scenarios, answering user queries depends on relevant clues scattered across different documents, necessitating cross-document information retrieval and integrated reasoning. However, RAG systems may struggle to fully grasp the complexity and diverse implicit information needs of such tasks, affecting the efficiency of the retrieval module and leading to the failure to identify all relevant cross-document segments.

3.2 Optimization Strategies

Building upon the diagnostic framework established in our failure taxonomy, this section details a corresponding suite of targeted optimization strategies. For each of the five identified failure modes, we propose specific techniques designed to mitigate its root cause. These strategies span various intervention points within the RAG pipeline, including pre-retrieval query manipulation, advanced retrieval mechanisms, and post-retrieval processing. The following subsections will elaborate on the mechanism of each strategy.

Ranking Confusion and Noise Impact. To mitigate ranking confusion, we investigate several advanced retrieval techniques. We employ a sophisticated **re-ranking** model (Cohere's rerank) to perform fine-grained relevance scoring on an initial set of retrieved documents. Furthermore, we implement **hybrid retrieval**, combining sparse (BM25) and dense retrieval methods to capture both lexical and semantic relevance. We also evaluate a combined pipeline that first uses hybrid retrieval to generate candidates and then applies re-ranking to refine the final order. The same **re-ranking** model is also leveraged to enhance robustness against noise impact. In this context, its purpose is twofold: not only to improve the order of relevant documents but also

(a) Golden Case (b) Negative Refusal (c) Ranking Confusion

(d) Answer Absence (e) Noise Impact (f) Complex Reasoning

Figure 1: Illustration of our proposed **RAG Failure Taxonomy**, providing concrete examples for the five key failure modes that form the basis of our diagnostic framework. Each case (b-f) highlights a specific breakdown in the RAG pipeline by contrasting the system's flawed output (Actual Response) against the ground truth (Expected Response). The 'Golden Case' (a) serves as a reference for successful RAG execution. A golden case alongside examples of typical RAG failures.

to actively filter out irrelevant or "noisy" documents before they are passed to the LLM.

Answer Absence. For the answer absence problem, we explore four context aggregation strategies provided by LlamaIndex(LlamaIndex Team, 2023) to investigate their impact on answer completeness. These are: (1) Simple Summarize, where all chunks are concatenated into a single prompt; (2) Refine, an iterative process where the LLM sequentially refines its answer with each new chunk; (3) Compact, which optimizes "Refine" by first merging chunks to reduce LLM calls; and (4) Compact Accumulate, where merged chunks are used for independent Q&A sessions, with the final answer being an aggregation of all results.

Negative Refusal. To address negative refusal, we test two strategies. The first is Prompt Engineering, which uses explicit instructions to encourage the model to refuse answering when appropriate. The second is a Two-step Reasoning process, where the LLM first assesses the sufficiency of the provided information before deciding whether to generate an answer.

Complex Reasoning. For complex reasoning, we explore three techniques. At the pre-retrieval stage, we test **Query Rewriting**, which uses an LLM to enrich the user's query for better retrieval guidance, and **Query Decomposition**, which breaks a complex question into simpler sub-questions. At the post-retrieval stage, we apply **Few-shot Prompting**, providing the LLM with in-context examples of complex reasoning to improve its generative capabilities.

4 Experiments

To empirically validate our framework, we designed a series of experiments, each targeting a specific failure mode. This section first outlines the common setup for all experiments and then details the settings and results for each failure category in separate subsections.

4.1 Experimental Setup

Datasets and Evaluation. Our experiments are based on the **Natural Questions (NQ)** (Kwiatkowski et al., 2019) and **HotpotQA** (Yang et al., 2018) datasets. Rather than wholesale evaluation, we adopt a targeted diagnostic approach, curating a small, focused evaluation set (typically 20 samples) for each of the five failure modes. This method is designed to reliably trigger and isolate specific failures, enabling a controlled and interpretable assessment of our mitigation strategies. For experiments involving LLM generation (Sections 4.3 through 4.5), we report the mean and standard deviation (mean $\pm$ std) over three runs with different random seeds to ensure reliability.

Our evaluation employs two sets of metrics. For retrieval quality, we use standard IR metrics: **F1, EM, MRR,** and **NDCG**. For generation quality, we utilize the **Uptrain** (Uptrain AI, 2023) framework for its LLM-as-a-judge evaluations, including: **Factual Accuracy (Up-FAcc)** (verifying grounding in context), **Response Completeness (Up-RCmp)** (addressing all parts of the query), and **Context Utilization (Up-CUti)** (evaluating reasoning over context vs. parametric recall).

Models and Implementation. Our RAG pipeline is implemented using the LlamaIndex framework (LlamaIndex Team, 2023) with the following components:

- **Generator:** We use gpt-3.5(Floridi and Chiriatti, 2020) as our primary generator LLM.
- **Retriever:** For dense retrieval, we employ the bge-large-en-v1.5 model (Zheng et al., 2023) and cosine similarity for top-5 search.
- **Re-ranker:** For all re-ranking tasks, we utilize Cohere's rerank-english-v2.0(Cohere, 2022) model.

The document corpus for both datasets is chunked with a size of 512 tokens and an overlap of 128 tokens. These standard parameters are chosen to ensure a fair and consistent basis for comparison across all evaluated strategies.

Results. As shown in Table 2, both strategies achieve a perfect rejection rate on wrong contexts. However, their reliability diverges significantly on correct contexts. The "Prompt Engineering" method is highly volatile, incorrectly refusing to answer 60% of the time with a large standard deviation, severely undermining its usability. In contrast, the "Two-step Reasoning" method is far more stable, exhibiting a much lower incorrect refusal rate of 10% with minimal variance, demonstrating its superior reliability.

4.2 Ranking Confusion

Settings. The diagnostic set for this failure was curated by a two-step filtering process on the NQ dataset: 1) We identified instances where the Basic-RAG system yielded a low F1 score (< 0.5). 2) From this pool, we manually verified that the correct answer document *was* present in the top-5 retrieved results. Each sample is thus a canonical example of generator failure due to poor ranking, not missing information.

Results. As shown in Table 5, system performance degrades as noise increases. Applying a re-ranking step consistently mitigates this. Notably, when three noisy documents are present, re-ranking not only improves key metrics like completeness (from 0.400 to 0.700) but also demonstrates superior stability. This is evidenced by its consistently lower standard deviations across all metrics compared to the baseline without re-ranking, confirming its role as a crucial stabilizer in noisy environments.

Table 1: A systematic framework for diagnosing and mitigating RAG failures. This table outlines the core components: a taxonomy of failure modes, a mapping to targeted optimization strategies, and the specific dataset settings and metrics used for evaluation. Strategies are categorized as pre-retrieval (♣), post-processing (▼), or advanced retriever (♠).

RAG Failures	Optimization Strategies	Dataset Settings	Evaluation Metrics
Negative Refusal	▼ Prompt Engineering ▼ Two-step Reasoning	Randomly sampled queries with unrelated context	Rejection Rate
Ranking Confusion	▼ Re-ranking ♠ Hybrid Retrieval ♠ ▼ Hybrid Retrieval & Re-ranking	Samples with lower F1 scores	F1, Hit@1, EM, MRR, MAP, DCG, IDCG, NDCG
Answer Absence	▼ Simple Summarize ▼ Refine ▼ Compact ▼ Compact Accumulate	Samples with missing answers	Factual Accuracy (Up-FAcc) Response Consistency (Up-RCns) Context Utilization (Up-CUti) Response Completeness (Up-RCmp) Response Matching (Up-RMch)
Noise Impact	▼ Re-ranking	Random samples with varying numbers of noisy context	
Complex Reasoning	♣ Query Rewriting ♣ Query Decomposition ▼ Few-shot Prompting	Random samples from HotpotQA Hard	

Table 2: Evaluation of negative refusal strategies.

Strategies		Rejection Rate
Context Type	Method	
Wrong Context	Prompt Engineering	1.000
	Two-step Reasoning	1.000
Correct Context	Prompt Engineering	0.600
	Two-step Reasoning	0.100

4.3 Answer Absence

Settings. This evaluation set was constructed by identifying 20 multi-hop questions from HotpotQA where the Basic-RAG system, despite being provided with all necessary golden context documents, produced an incomplete answer missing at least one key entity or relation. This methodology precisely isolates the failure to the generator's inability to synthesize information from multiple provided sources, rather than a retrieval error.

Results. Table 4 reveals that different context aggregation strategies significantly impact answer completeness. The "Refine" strategy achieves the highest factual accuracy (0.940), but at the cost of lower completeness. Conversely, "Compact Accumulate" not only achieves high completeness (0.850) but also the best context utilization (0.975) with minimal performance variance (±0.004), indicating its reliability.

4.4 Noise Impact

Settings. To simulate real-world scenarios where retrieval is imperfect and often includes noisy or irrelevant results, we created a diagnostic set from HotpotQA. For each of the 20 questions, we started with the golden context documents and systematically injected 1, 2, or 3 "distractor" documents retrieved for an unrelated question. This creates a controlled, titratable environment to rigorously measure the system's resilience against contextual noise.

Results. As shown in Table 5, while system performance degrades with increasing noise, applying a re-ranking step consistently mitigates this. This is most pronounced with three noisy documents, where re-ranking significantly improves response completeness from 0.400 to 0.700. Crucially, this gain is accompanied by greater stability, evidenced by its consistently lower standard deviations, confirming its role as a vital defensive mechanism against contextual noise.

4.5 Complex Reasoning

Settings. The testbed for complex reasoning was curated from the 'Hard' subset of

Table 3: Results for ranking confusion strategies (RR: Re-ranking, HR: Hybrid Retrieval). Since these are deterministic retrieval metrics based on embeddings, they do not have standard deviations across runs. The up arrow (↑) indicates that higher values are better.

Strategy	**F1 (↑)**	**EM (↑)**	**MRR (↑)**	**HIT@1 (↑)**	**MAP (↑)**	**DCG (↑)**	**IDCG (↑)**	**NDCG (↑)**
Basic-RAG	0.740	0.000	0.667	0.900	0.680	1.2100	1.5300	0.780
w/ RR	0.800	0.000	0.750	**1.000**	0.792	**1.6310**	1.3810	0.847
w/ HR	0.925	0.850	0.925	0.900	0.913	1.4800	1.5360	0.945
w/ HR + RR	**0.975**	**0.950**	**0.975**	**1.000**	**1.000**	1.6000	1.6310	**0.982**

Table 4: Evaluation of answer absence strategies. We report mean ± std over 3 runs. The up arrow (↑) indicates that higher values are better.

	Evaluation Metrics				
Strategy	**Up-FAcc (↑)**	**Up-RCns (↑)**	**Up-CUti (↑)**	**Up-RCmp (↑)**	**Up-RMch (↑)**
Simple Summarize	0.025 ± 0.005	**1.000**	0.825 ± 0.012	$\mathbf{0.850 \pm 0.009}$	0.279 ± 0.026
Refine	$\mathbf{0.940 \pm 0.006}$	**1.000**	0.675 ± 0.018	0.575 ± 0.021	0.205 ± 0.033
Compact	0.875 ± 0.009	**1.000**	0.700 ± 0.015	0.425 ± 0.024	0.293 ± 0.029
Compact Accumulate	0.819 ± 0.011	0.965 ± 0.007	$\mathbf{0.975 \pm 0.004}$	$\mathbf{0.850 \pm 0.008}$	$\mathbf{0.384 \pm 0.014}$

HotpotQA, which is specifically designed for multi-hop reasoning. We selected 20 questions that necessitate synthesizing information from at least two different documents to formulate a correct answer, representing a challenging benchmark for reasoning capabilities.

Results. The results in Table 6 demonstrate that query rewriting provides a significant boost, improving factual accuracy from 0.570 to 0.675. Crucially, its low standard deviation (±0.008) indicates this improvement is highly consistent and reliable. In stark contrast, both query decomposition and few-shot prompting led to a decline in performance with high variance (e.g., std of ±0.035 and ±0.041 on Up-CUti), revealing their unreliability and unpredictable behavior on such tasks.

5 Analysis

Our experimental results not only quantify the performance gains of the proposed strategies but also reveal several cross-cutting principles crucial for designing robust and effective RAG systems. We synthesize our findings into three key insights.

5.1 The Perils of Unchecked Complexity

A recurring theme across our experiments is that simply increasing pipeline complexity is not a panacea and can even be detrimental. For instance, in the Answer Absence task (Table 4), the complex "Refine" strategy achieves the highest factual accuracy (0.940) but at a significant cost to response completeness (0.575) and introduces instability (±0.021). In contrast, the simpler "Compact Accumulate" strategy strikes a better balance with high completeness (0.850) and strong accuracy (0.819) at very low variance. This is reinforced by our Complex Reasoning results (Table 6), where sophisticated strategies like "Query Decomposition" proved unpredictable, while the simpler, targeted "Query Rewriting" was demonstrably more stable and effective. The takeaway is that interventions should be purposeful and targeted, as layering on complexity can introduce performance variance and reduce overall robustness.

5.2 Retrieval Quality as the Cornerstone of Robustness

Our findings consistently underscore that retrieval quality is the most critical factor for end-to-end system performance. In the Ranking Confusion task (Table 3), combining hybrid retrieval with re-ranking (HR + RR) yielded the most dramatic gains, achieving a near-perfect NDCG of 0.982. This highlights a powerful synergy: hybrid retrieval broadens the candidate pool, while re-ranking precisely filters for relevance. This principle is further demonstrated in the Noise Impact experiment (Table 5), where re-ranking consistently miti-

Table 5: Evaluation of noise impact strategies. We report mean $\pm$ std over 3 runs. The up arrow (↑) indicates that higher values are better.

		Evaluation Metrics				
Noise Number	**Strategy**	**Up-FAcc (↑)**	**Up-RCns (↑)**	**Up-CUti (↑)**	**Up-RCmp (↑)**	**Up-RMch (↑)**
0	w/o Re-ranking	0.895 ± 0.004	1.000	0.675 ± 0.006	0.500 ± 0.008	0.392 ± 0.011
1	w/o Re-ranking	$\mathbf{0.900 \pm 0.005}$	$\mathbf{0.995 \pm 0.002}$	0.800 ± 0.011	0.450 ± 0.025	$\mathbf{0.470 \pm 0.018}$
	w/ Re-ranking	0.679 ± 0.018	0.798 ± 0.021	$\mathbf{0.857 \pm 0.009}$	$\mathbf{0.571 \pm 0.014}$	0.236 ± 0.031
2	w/o Re-ranking	$\mathbf{0.865 \pm 0.008}$	0.950 ± 0.015	$\mathbf{0.750 \pm 0.013}$	0.400 ± 0.028	0.376 ± 0.025
	w/ Re-ranking	0.588 ± 0.022	**1.000**	0.650 ± 0.019	$\mathbf{0.550 \pm 0.016}$	$\mathbf{0.400 \pm 0.021}$
3	w/o Re-ranking	0.793 ± 0.015	0.950 ± 0.014	$\mathbf{0.700 \pm 0.011}$	0.400 ± 0.031	0.298 ± 0.029
	w/ Re-ranking	$\mathbf{0.802 \pm 0.009}$	**1.000**	$\mathbf{0.700 \pm 0.011}$	$\mathbf{0.700 \pm 0.012}$	$\mathbf{0.480 \pm 0.016}$

Table 6: Evaluation of complex reasoning strategies. We report mean $\pm$ std over 3 runs. The up arrow (↑) indicates that higher values are better.

	Evaluation Metrics				
Strategy	**Up-FAcc (↑)**	**Up-RCns (↑)**	**Up-CUti (↑)**	**Up-RCmp (↑)**	**Up-RMch (↑)**
Basic-RAG	0.570 ± 0.022	**1.000**	0.750 ± 0.018	0.400 ± 0.029	0.383 ± 0.025
Query Rewriting	$\mathbf{0.675 \pm 0.008}$	0.990 ± 0.004	$\mathbf{0.775 \pm 0.009}$	$\mathbf{0.550 \pm 0.011}$	$\mathbf{0.443 \pm 0.014}$
Query Decomposition	0.625 ± 0.031	0.950 ± 0.024	0.550 ± 0.035	0.325 ± 0.038	0.279 ± 0.033
Few-shot Prompting	0.625 ± 0.029	0.890 ± 0.028	0.525 ± 0.041	0.250 ± 0.043	0.241 ± 0.038

gated performance degradation and variance, especially in high-noise scenarios (e.g., boosting completeness from 0.400 to 0.700). This confirms that a strong re-ranking component acts as a vital defensive mechanism, enhancing the system's overall robustness and predictability.

5.3 Dynamic Safeguards Outperform Static Rules

The Negative Refusal experiment reveals a crucial distinction between static and dynamic safety mechanisms. While a static rule ("Prompt Engineering") proves brittle and unreliable on correct contexts, the dynamic "Two-step Reasoning" method achieves perfect refusal on wrong contexts without compromising performance on correct ones. Its superior reliability stems from its ability to first explicitly reason about context relevance before committing to an answer, providing the flexibility and consistency that a rigid prompt lacks. This highlights a key design principle: critical safety features should be implemented as dynamic, context-aware processes to ensure both effectiveness and reliability.

6 Conclusion

To address the challenge of subtle yet critical failures in RAG systems, we introduced a comprehensive framework centered on a detailed **RAG Failure Taxonomy** and a corresponding suite of empirically validated **Optimization Strategies**.

Our experiments yielded crucial insights for building robust systems, demonstrating that targeted interventions often outperform unchecked complexity, that retrieval quality is the cornerstone of end-to-end performance, and that dynamic, context-aware safeguards are significantly more reliable than static rules.

Ultimately, our work provides a principled methodology that transforms RAG optimization from an ad-hoc, trial-and-error exercise into a more structured engineering discipline. Future work could extend this framework to multi-modal RAG applications and investigate the interaction effects between co-occurring failures.

7 Limitations

While our study provides a systematic framework for diagnosing and mitigating common RAG failures, we acknowledge several limitations that offer avenues for future research.

Scope of the Taxonomy. Our proposed taxonomy categorizes five prevalent failure modes observed in text-based question-answering scenarios. However, this taxonomy is not exhaustive. RAG systems deployed in specialized domains or handling multi-modal data (e.g., tables, images) may exhibit different or more nuanced failure modes not covered by our current framework. Future work could extend this taxonomy to encompass a broader range of data types and application contexts.

Analysis of Failures in Isolation. To ensure a clear and controlled analysis, our experiments were designed to isolate each of the five failure modes. For example, the *Noise Impact* experiments involved synthetically injecting a controlled number of irrelevant documents. In real-world applications, however, these failures often co-occur and interact. A complex query (*Complex Reasoning*) might retrieve documents where the correct answer is buried in the middle (*Ranking Confusion*) amidst several noisy passages (*Noise Impact*). Our current study does not investigate these complex interaction effects. A valuable direction for future work is to study how these failures compound and whether a sequence or combination of mitigation strategies is more effective in such multifaceted scenarios.

Despite these limitations, we believe our work provides a foundational and actionable methodology that moves the process of RAG optimization from ad-hoc trial-and-error towards a more structured and principled engineering discipline.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Manoj Ghuhan Arivazhagan, Lan Liu, Peng Qi, Xinchi Chen, William Yang Wang, and Zhiheng Huang. 2023. Hybrid hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10680–10689.

Akari Asai, Zeqiu Wu, Yizhong Santhanam, Wenhao Wang, Udi Alon, Mor Geva, Maartje Ganseman, Frank F. Chi, Wen-tau Yih, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, and 1 others. 2023. Benchmarking foundation models with language-model-as-an-examiner. *Advances in Neural Information Processing Systems*, 36:78142–78167.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087.

Cohere. 2022. Introducing cohere rerank 2.0. https://txt.cohere.com/rerank-2/. Accessed: 2025-10-04.

Andrei C Coman, Ionut-Teodor Sorodoc, Leonardo FR Ribeiro, Bill Byrne, James Henderson, and Adrià de Gispert. 2025. Ragferee: Building contextual reward models for retrieval-augmented generation. *arXiv preprint arXiv:2509.26011*.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158.

Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds and machines*, 30(4):681–694.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. 2025. Deeprag: Thinking to retrieve step by step for large language models. *arXiv preprint arXiv:2502.01142*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Junyi Li and Hwee Tou Ng. 2024. Think&cite: Improving attributed text generation with self-guided tree search and progress reward modeling. *arXiv preprint arXiv:2412.14860*.

Kevin Lin, Kyle Lo, Joseph E Gonzalez, and Dan Klein. 2023. Decomposing complex queries for tip-of-the-tongue retrieval. *arXiv preprint arXiv:2305.15053*.

Haowei Liu, Xuyang Wu, Guohao Sun, Zhiqiang Tao, and Yi Fang. 2024. Ract: Ranking-aware chain-of-thought optimization for llms. *arXiv preprint arXiv:2412.14405*.

Nelson F. Liu, Kevin Lin, John Hewitt, Alexandra Cheung, Percy Liu, and Christopher D. Manning. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

LlamaIndex Team. 2023. Llamaindex: A data framework for llm applications. https://github.com/run-llama/llama_index. Accessed on 2025-10-04.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Leopold Müller, Joshua Holstein, Sarah Bause, Gerhard Satzger, and Niklas Kühl. 2025. Data quality challenges in retrieval-augmented generation. *arXiv preprint arXiv:2510.00552*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Guodong Long, Kai Zhang, and Daxin Jiang. 2023. Unifier: A unified retriever for large-scale retrieval. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4787–4799.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*.

Paras Nath Singh, Sreya Talasila, and Shivaraj Veerappa Banakar. 2023. Analyzing embedding models for embedding vectors in vector databases. In *2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG)*, pages 1–7. IEEE.

Shangeetha Sivasothy, Scott Barnett, Stefanus Kurniawan, Zafaryab Rasool, and Rajesh Vasa. 2025. Ragprobe: Breaking rag pipelines with evaluation scenarios. In *2025 IEEE/ACM 4th International Conference on AI Engineering–Software Engineering for AI (CAIN)*, pages 60–71. IEEE.

Uptrain AI. 2023. Uptrain: Open-source llm and rag evaluation framework. https://uptrain.ai/. Accessed on 2025-10-04.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Gui, Ziran Jiang, Ziyu Jiang, and 1 others. 2024. Crag-comprehensive rag benchmark. *Advances in Neural Information Processing Systems*, 37:10470–10490.

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

Ningning Zhang, Chi Zhang, Zhizhong Tan, Xingxing Yang, Weiping Deng, and Wenyong Wang. 2025a. Credible plan-driven rag method for multi-hop question answering. *arXiv preprint arXiv:2504.16787*.

Xiangrui Zhang, Fuyong Zhao, Yutian Liu, Panfeng Chen, Yanhao Wang, Xiaohua Wang, Dan Ma, Huarong Xu, Mei Chen, and Hui Li. 2025b. Treeqa: Enhanced llm-rag with logic tree reasoning for reliable and interpretable multi-hop question answering. *Knowledge-Based Systems*, page 114526.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

A Prompt for XRAG Instructions

Prompt for XRAG suitable for HotpotQA datasets

Context information is below.
. . . . . .
{context_str}
. . . . . .
Given the context information and no prior knowledge, answer the question:
{query_str}
We have the opportunity to refine the original answer.
(only if needed) with some more context below.
. . . . . .
{context_msg}
. . . . . .
Given the new context, refine the original answer to better.
Answer the question: query_str
If the context isn't proper, output the original answer again.
Original Answer: existing_answer

Prompt for XRAG suitable for DropQA and NaturalQA datasets

Context information is below.
. . . . . .
{context_str}
. . . . . .
Given the context information and no prior knowledge. Please provide a brief, shortest possible answer, ideally just one word for the following question:
Question: who has sold more records Oasis or Coldplay?
Expected Answer: Oasis
We have the opportunity to refine the original answer.
(only if needed) with some more context below.
. . . . . .
{context_msg}
. . . . . .
Given the new context, refine the original answer to better.
Answer the question: query_str
If the context isn't proper, output the original answer again.
Original Answer: existing_answer

B Case Studies of RAG Failures

To further illustrate the failure modes defined in our taxonomy (Section 3.1), this section provides concrete examples for each category. Each case study includes the user query, the context provided to the generator, the system's erroneous output, and an analysis of the failure.

B.1 Case Study: Negative Refusal

Failure Mode: Negative Refusal

Query: "What was the official mascot for the 2025 Winter Olympics?"

Retrieved Context:

> [Doc 1] The 2022 Winter Olympics, held in Beijing, featured the mascot Bing Dwen Dwen, a panda in an ice suit...
> [Doc 2] The tradition of Olympic mascots began in 1968. They are designed to represent the culture of the host country...

System's (Incorrect) Output: "The official mascot for the 2025 Winter Olympics was a snow leopard named 'Leo'."

Analysis: *The retrieved context contains information about past mascots but nothing about the 2025 Games. Instead of acknowledging the absence of information, the model hallucinates a plausible-sounding but entirely fabricated answer. A robust system should have refused to answer, stating that the information is not available.*

B.2 Case Study: Ranking Confusion

Failure Mode: Ranking Confusion

Query: "Who holds the record for the most Grand Slam singles titles in men's tennis?"

Retrieved Context (Top 5):

> [Doc 1] Roger Federer's emotional retirement in 2022 marked the end of an era. He won 20 Grand Slam titles...
> [Doc 2] Rafael Nadal, the king of clay, has battled injuries throughout his career. He has won 22 Grand Slam titles...
> [Doc 3] Pete Sampras was a dominant force in the 1990s, retiring with 14 Grand Slam titles...
> [Doc 4] As of late 2024, Novak Djokovic holds the all-time record for men's singles Grand Slam titles, with 24 championships.
> [Doc 5] The rivalry between Federer, Nadal, and Djokovic is considered the greatest in tennis history...

System's (Incorrect) Output: "Rafael Nadal holds the record for the most Grand Slam singles titles with 22."

Analysis: *The correct answer is present in Doc 4, but it is "lost in the middle" of the retrieved list. The model incorrectly focuses on the information presented in the top-ranked documents (Doc 1 and 2), leading to an outdated and factually wrong answer. This exemplifies the "lost-in-the-middle" problem.*

B.3 Case Study: Answer Absence

Failure Mode: Answer Absence

Query: "What are the main functions of the liver and where is it located in the human body?"

Retrieved Context:

> [Doc 1] The liver is a vital organ located in the upper right quadrant of the abdomen, beneath the diaphragm. It performs over 500 essential tasks. Key functions include detoxifying chemicals, metabolizing drugs, and producing bile...

System's (Incomplete) Output: "The liver's main functions include detoxifying chemicals, metabolizing drugs, and producing bile."

Analysis: *The retrieved context (Doc 1) contains all the necessary information to answer the query completely (both functions and location). However, the model's generated response only includes the functions and completely omits the location, resulting in an incomplete answer.*

B.4 Case Study: Noise Impact

Failure Mode: Noise Impact

Query: "What is the primary ingredient in traditional Japanese miso soup?"

Retrieved Context:

> [Doc 1] (Noise) Dashi is a fundamental stock in Japanese cooking, often made from kombu (kelp) and bonito flakes...
> [Doc 2] Miso soup is a traditional Japanese soup consisting of a stock called 'dashi' into which softened miso paste is mixed...
> [Doc 3] (Noise) Soy sauce is another key ingredient in Japanese cuisine, used in everything from sushi to teriyaki...

System's (Incorrect) Output: "The primary ingredient in traditional Japanese miso soup is dashi, which is made from kelp and bonito flakes."

Analysis: *Although the correct document (Doc 2) was retrieved, the presence of noisy but related documents (Doc 1 about dashi, Doc 3 about soy sauce) confused the model. It incorrectly identified dashi (the stock) as the primary ingredient instead of miso paste, which is the defining component mentioned in Doc 2.*

B.5 Case Study: Complex Reasoning

Failure Mode: Complex Reasoning

Query: "Who is the CEO of the company that developed the game Elden Ring?"

Retrieved Context:

> [Doc 1] Elden Ring is an action role-playing game developed by FromSoftware and published by Bandai Namco Entertainment...
> [Doc 2] Hidetaka Miyazaki is the current president and CEO of FromSoftware. He is also a renowned game director, known for the Souls series...

System's (Incorrect) Output: "Elden Ring was developed by the company FromSoftware."

Analysis: *This multi-hop question requires a two-step reasoning process: (1) Find the company that developed Elden Ring (FromSoftware, from Doc 1). (2) Find the CEO of that company (Hidetaka Miyazaki, from Doc 2). The system successfully retrieved both necessary pieces of information but failed to synthesize them. It only completed the first step of the reasoning chain, providing an answer that is correct but does not fully address the user's query.*

C Prompts for Complex Reasoning Strategies

This section provides the detailed prompts used to implement the complex reasoning strategies evaluated in our experiments (Table 6). Each prompt is designed to guide the LLM in performing a specific, targeted reasoning task to handle complex user queries.

Prompt for Query Rewriting

You are an expert query rewriter. Your task is to transform a given user query, which may be ambiguous or part of a conversation, into a clear, self-contained question optimized for a vector database search. The rewritten query should be specific and resolve any dependencies on prior context.

Original Query: {query}

Rewritten Query:

Prompt for Query Decomposition

You are a helpful assistant that decomposes complex questions into a series of simpler, interconnected sub-questions. Given a complex user question, break it down into a step-by-step sequence of questions that can be answered individually to build up to the final answer.

Example:

> **Complex Question:** "Which film, directed by the same person who directed 'Titanic', won the Academy Award for Best Picture?"
>
> **Decomposed Questions:**
>
> 1. Who directed the film 'Titanic'?
> 2. Which films directed by [answer to 1] have won the Academy Award for Best Picture?

Complex Question: {query}

Decomposed Questions:

Structure for Few-shot Prompting

The Few-shot Prompting strategy augments the generator's system prompt with several examples ("shots") of high-quality reasoning. This guides the model to follow a similar reasoning pattern. Below is a representative example of a single "shot" used in our experiments.

> **Question:** Which two U.S. presidents who served in the 21st century were born in the same state?
>
> **Context:** ... George W. Bush, the 43rd U.S. President, served from 2001 to 2009. He was born in New Haven, Connecticut... Barack Obama, the 44th U.S. President, served from 2009 to 2017. He was born in Honolulu, Hawaii...
>
> **Answer (with reasoning):** *Based on the provided context, two U.S. presidents who served in the 21st century were born in different states. George W. Bush, who served from 2001-2009, was born in Connecticut. Barack Obama, who served from 2009-2017, was born in Hawaii. Therefore, based on the context, there are no two presidents fitting the criteria who were born in the same state.*