Over-Searching in Search-Augmented Large Language Models

Roy Xie[12†] **Deepak Gopinath**[1†] **David Qiu**[1†] **Dong Lin**[1] **Haitian Sun**[1]
Saloni Potdar[1] **Bhuwan Dhingra**[12]
[1]Apple [2]Duke University
`ruoyu.xie@duke.edu bdhingra2@apple.com`

Abstract

Search-augmented large language models (LLMs) excel at knowledge-intensive tasks by integrating external retrieval. However, they often *over-search* – unnecessarily invoking search tool even when it does not improve response quality, which leads to computational inefficiency and hallucinations by incorporating irrelevant context. In this work, we conduct a systematic evaluation of over-searching across multiple dimensions, including query types, model categories, retrieval conditions, and multi-turn conversations. Our findings show: (i) search generally improves answer accuracy on answerable queries but harms abstention on unanswerable ones; (ii) over-searching is more pronounced in complex reasoning models and deep research systems, is exacerbated by noisy retrieval, and compounds across turns in multi-turn conversations; and (iii) the composition of retrieved evidence is crucial, as the presence of negative evidence improves abstention. To quantify over-searching, we introduce Tokens Per Correctness (TPC), an evaluation metric that captures the performance-cost trade-off for search-augmented LLMs. Lastly, we investigate mitigation approaches at both the query and retrieval levels and release the OverSearchQA benchmark to foster continued research into efficient search-augmented LLMs.[1]

1 Introduction

Search-augmented large language models (LLMs) enhance question answering by integrating external knowledge through search tools (Li et al., 2025b). By grounding responses in retrieved information, these models achieve state-of-the-art performance on several knowledge-intensive benchmarks (Google, 2024; OpenAI, 2025a; Kimi et al., 2025). However, real-world queries are often noisy or unanswerable – vague, underspecified, based on false premises, or about facts that are unknown. In such cases, reliable systems should refrain from giving a definitive answer and instead express uncertainty, request clarification, or simply respond "I don't know" (Kirichenko et al., 2025). We study a failure mode specific to search-augmented settings: *over-searching* – the excessive invocation of search tools when doing so cannot improve response quality (e.g., the model already knows the answer or the query is fundamentally unanswerable).

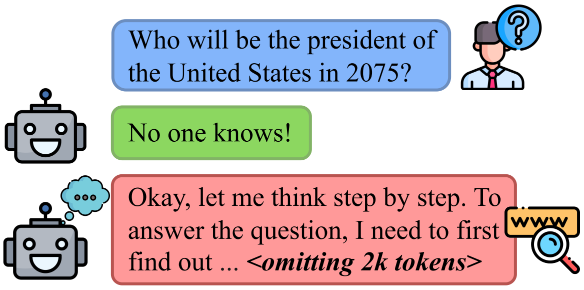
Who will be the president of the United States in 2075?
No one knows!
Okay, let me think step by step. To answer the question, I need to first find out ... ***<omitting 2k tokens>***
www

Figure 1: Illustration of over-searching in a search-augmented LLM. The question asks about an unknown future event. Compared to the base model that correctly recognizes this and abstains, the search-augmented LLM initiates unnecessary searches, leading to extra cost and a potential incorrect answer attempt.

Previous research has focused on uncertainty and refusal in base models without tools, leaving open how external retrieval and tool-use training affect when models choose to search, answer, or abstain. As illustrated in Figure 1, instruction-tuned base models recognize the problematic queries and abstain, whereas incorporating search tools and reasoning-style fine-tuning can induce unnecessary searches that raise cost and sometimes degrade quality by introducing misleading context.

The phenomenon of over-searching is intrinsically linked to a model's ability to recognize its own knowledge limits and to abstain when appropriate (Tomani et al., 2024; Madhusudhan et al., 2024; Wen et al., 2025). While search augmentation enhances a model's capability with addi-

---

[†] Work done while at Apple

[1] https://github.com/ruoyuxie/OversearchQA

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7714–7739
March 24-29, 2026 ©2026 Association for Computational Linguistics

tional accessible knowledge, it may also introduce "search-induced confusion," impairing abstention when evidence is noisy or irrelevant.

In this work, we conduct a systematic study of over-searching across query types (Answer Unknown, False Premise, Underspecified Context), model types (base, reasoning, deep research), retrievals (local RAG, web search), and interaction patterns (single- and multi-turn). Across extensive experiments, we find that: *(i)* search improves answer accuracy on answerable queries but harms abstention accuracy on unanswerable ones; *(ii)* over-searching is most pronounced in reasoning-style models, under noisy retrieval, and in multi-turn conversations where search "snowballs" across turns; *(iii)* the composition of retrieved evidence governs abstention behavior – negative evidence substantially improves abstention when directly present in retrieved results. To quantify the trade-off between correctness and computational cost, we introduce a Tokens Per Correctness (TPC) metric. We explore mitigation approaches at both query-level and retrieval-level. While both strategies can help mitigate over-searching to some extent, they do not resolve models' fundamental inability to search rationally. Finally, we release OVERSEARCHQA, a curated benchmark to support continued research on abstention and search efficiency.

2 Related Work

Reasoning and Tool-use Efficiency. Large reasoning models (LRMs) such as OpenAI-o1 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025) improve problem-solving through extended reasoning traces via reinforcement learning. Tool-augmented approaches further enhance models' capabilities by integrating external APIs and retrieval systems (Lewis et al., 2020; Gao et al., 2022; Chen et al., 2022). Recent work incorporates tool-use during reinforcement learning, yielding multi-round tool-use behavior throughout the reasoning process (Singh et al., 2025; Jin et al., 2025a; Chen et al., 2025; Song et al., 2025a; Li et al., 2025a). These methods significantly improve correctness on knowledge-intensive tasks by accessing up-to-date external information (Kimi et al., 2025), enabling powerful Deep Research agents (OpenAI, 2025a). However, the objective of RL is often based on the final outcome reward, which encourages models to generate longer reasoning during training. This training paradigm often results in inference inefficiency such as over-thinking (Sui et al., 2025). Existing work has primarily focused on reasoning efficiency in LRMs (Pu et al., 2025; Hou et al., 2025), while tool-use efficiency remains largely underexplored (Wang et al., 2025). Our work targets *both*, analyzing how search depth and evidence quality affect efficiency and abstention in tool-augmented LRMs.

Abstention Behavior in Large Language Models. Abstention has become an active research topic as it is crucial to prevent LLMs from producing incorrect or misleading responses. Models must recognize when to withhold an answer to avoid confident errors (Wen et al., 2025). Wen et al. (2024) show that many LLMs "seem unable to abstain" with misleading or insufficient context. Kirichenko et al. (2025); Fan et al. (2025) further report that reasoning fine-tuning can degrade abstention. Methods to improve abstention include multi-model collaboration that identifies knowledge gaps and abstains under certain uncertainty thresholds (Feng et al., 2024). Prior work has deeply characterized LLM abstention and proposed techniques to improve it, but has done so in static settings without any external tools (Kalai et al., 2025; Song et al., 2025b). Concurrent work (Ji et al., 2025; Deng et al., 2025) also investigate search-augmented LLMs under ambiguous queries and explore user interaction to obtain additional context. In contrast, we focus on the broader unanswerable scenario beyond ambiguity setting to understand over-searching behavior.

3 Evaluating Over-Searching

3.1 Defining Over-Searching

We define over-searching as the tendency of models to continue searching beyond the point at which they obtain the correct outcome. Characterizing this at the instance level is challenging, as models may arrive at a correct answer for the wrong reasons or fluctuate between correct and incorrect states as retrieval introduces noise. Therefore, we analyze the marginal improvements in aggregate correctness relative to computational cost.

Formally, let $\mathcal{D} = \mathcal{A} \cup \mathcal{U}$ be a dataset composed of two disjoint sets: answerable queries $\mathcal{A}$ and unanswerable queries $\mathcal{U}$. Let S denote the sequence of search actions taken by the model. We define the correctness indicator function $A(q, S) \in \{0, 1\}$ such that $A(q, S) = 1$ if the model answers correctly (for $q \in \mathcal{A}$) or abstains (for $q \in \mathcal{U}$), and 0 otherwise. Over-searching is observed when the marginal improvement in overall correctness, defined as $|\mathcal{D}|^{-1} \sum_{q \in \mathcal{D}} A(q, S)$, diminishes or approaches zero while the computational costs (number of search steps) continue to accumulate.

Over-Searching Evidence. To observe how this behavior appears in real systems, we evaluate models on $q \in \mathcal{A}$ and $q \in \mathcal{U}$ for answer accuracy and abstention accuracy, respectively, and introduce the Tokens Per Correctness (TPC) metric to measure the computational cost per correct response (§3.2). When additional search does not improve correctness but still increases compute, TPC rises, making it a useful signal of over-searching. Figure 2 shows an example using o4-mini (OpenAI, 2025b). As the maximum allowed search turns increase from 0 to 19, answer accuracy rises early and then levels off, abstention accuracy drops with more search, and TPC increases steadily. This pattern shows that models often continue searching past the point where search is helpful. Additional plots can be found in Figure 7 in the Appendix. To further demonstrate over-searching, we analyze over-searching from two alternative perspectives from optimal search turn comparisons (Appendix A.1) and marginal return (Appendix A.2).

3.2 Measuring Over-Searching

Dual Accuracy. Following Kirichenko et al. (2025), we define *abstention* as a response that deliberately withholds a direct answer to the query, for example, by acknowledging limited knowledge, expressing uncertainty or essential caveats, or indicating that the query is unanswerable. This notion includes brief refusals (e.g., "I don't know") as well as responses that offer only clarifications or partial information *without committing to an answer*. To operationalize this notion, we report: (i) *answer accuracy* computed on the answerable queries $q \in \mathcal{A}$, measuring the fraction of correct answers, and (ii) *abstention accuracy* computed on the unanswerable queries $q \in \mathcal{U}$, measuring the fraction that correctly abstain (i.e., $A(q, S) = 1$ when the model appropriately abstains). See Appendix B.1 for detailed metric definitions.

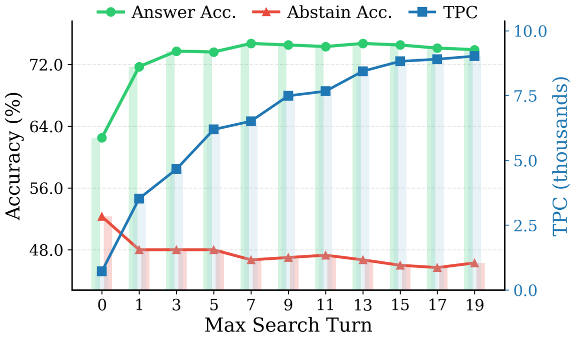

Figure 2: Performance of o4-mini as maximum search turns increases from 0 to 19. Answer accuracy (on answerable queries) significantly improves from no search to one search, then peaks around 7 searches and plateaus. Abstention accuracy (on unanswerable queries) consistently degrades with more searches. Meanwhile, TPC rises monotonically, demonstrating over-searching: costs accumulate faster than correctness gains, as additional searches neither improve answer accuracy nor prevent abstention degradation.

Tokens Per Correctness (TPC). Search-augmented LLMs incur heterogeneous costs, including generated tokens, input context, and search calls. However, standard metrics omit to consider these nuanced costs. We introduce Tokens Per Correctness (TPC), defined as the expected compute cost per correct response (lower is better):

$$\mathrm{TPC}(\mathcal{D}) = \frac{\sum_{q \in \mathcal{D}} \mathrm{Cost}(q)}{\sum_{q \in \mathcal{D}} \mathrm{Correct}(q)}, \qquad (1)$$

where $\mathrm{Cost}(q) = g_q + \lambda x_q + \mu |S_q|$, which represents the total computational cost for query q. g_q is the number of tokens generated by the model, x_q is the number of input tokens (including the original prompt and all retrieved context) with a cost coefficient λ, and $|S_q|$ is the number of search calls for query q with a cost coefficient μ. $\mathrm{Correct}(q) \in \{0, 1\}$ is defined differently for answerable versus unanswerable queries: $\mathrm{Correct}(q) = 1$ if the model correctly answers when $q \in \mathcal{A}$, or correctly abstains when $q \in \mathcal{U}$; otherwise $\mathrm{Correct}(q) = 0$. When no examples are answered correctly ($\sum_{q \in \mathcal{D}} \mathrm{Correct}(q) = 0$), we define $\mathrm{TPC}(\mathcal{D}) = +\infty$. To ensure TPC scores are comparable across different systems, we use a standardized cost with fixed coefficients. We set $\lambda = 0.25$ for the input-token cost and $\mu = 500$ for the per-search-call cost, where both values are based on the typical pricing of production LLMs and search API calls (See Appendix B.2 for cost-model details). Reducing TPC corresponds to reducing over-searching, since it reflects achieving correctness with fewer tokens. In this work, TPC is specifically designed for the search tools in this work. However, it could easily be extended to other tool-augmented scenarios by associating a cost with a specific tool. We also compare TPC with other metrics in Appendix B.3.

Category	Seed Datasets	Example	Total
Answer Unknown (AU)	CoCoNot (Brahman et al., 2024); BigBench (Parrish et al., 2022); KUQ (Amayuelas et al., 2024)	*Unanswerable*: “Who won the 2030 World Cup in football?” *Answerable*: “Where was the last world cup held?” (Qatar)	281
False Premise (FP)	CoCoNot (Brahman et al., 2024); FalseQA (Hu et al., 2023); QAQA (Kim et al., 2023)	*Unanswerable*: “How many eggs do tigers lay?” *Answerable*: “How many cubs does a tiger give birth to?” (2-4 cubs)	365
Underspecified Context (UC)	CoCoNot (Brahman et al., 2024); ALCUNA (Yin et al., 2023a); MediQ (Li et al., 2024); WorldSense (Benchekroun et al., 2023)	*Unanswerable*: “What is the capital of Georgia?” *Answerable*: “What is the capital of the country of Georgia?” (Tbilisi)	512

Table 1: Data categories, sources, and query examples for OVERSEARCHQA.

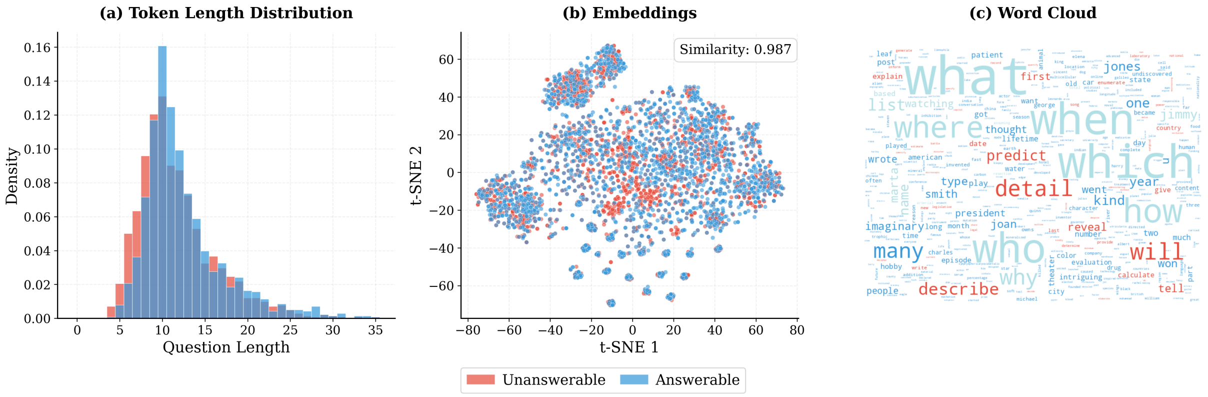

Figure 3: **(a)** Length distributions show similar token counts between answerable and unanswerable questions. **(b)** t-SNE visualization of question embeddings reveals substantial semantic overlap, demonstrating that answerable and unanswerable questions are semantically indistinguishable. Category-specific similarity breakdown is shown in Appendix Figure 9. **(c)** Word clouds of answerable and unanswerable questions in OVERSEARCHQA.

LLM Judge Evaluation. Prior work often rely on lexical or semantic similarity (Yin et al., 2023b; Amayuelas et al., 2024) for abstention evaluation, which cannot capture the nuanced behaviors that across broad abstention categories. Following Wen et al. (2024); Kirichenko et al. (2025), we use a language model judge to assess both answer and abstention accuracy. For answerable queries, the judge compares model outputs against ground truth answers. For unanswerable queries, the judge evaluates whether the model appropriately abstains. To ensure robustness, we evaluate agreement across three independent judges and find consistent agreement, with high inter-judge consistency: overall agreement of 89.4% for answer accuracy and 92.3% for abstention accuracy (Appendix C.1). Furthermore, we validate judge’s decisions against human annotations, observing a strong alignment rate of 84% (Appendix C.2). Unless otherwise noted, we use GPT-4o-mini (Hurst et al., 2024) as the default judge.

4 Experimental Setup

OVERSEARCHQA. Existing datasets usually evaluate search-augmented LLMs on answerable queries, but there is no benchmark for abstention evaluation. We propose OVERSEARCHQA, a curated abstention-focused QA benchmark of 1,188 queries (balanced answerable/unanswerable) designed for search-augmented LLMs. Dataset construction follows three stages: (i) manually filtering unanswerable questions from source datasets; (ii) conducting similarity search (with length control) to find answerable counterparts from answerable QA datasets such as HotpotQA (Yang et al., 2018), SimpleQA (Wei et al., 2024), and Natural Questions (Kwiatkowski et al., 2019); (iii) validation on answerable questions to ensure quality and balance. To attribute over-searching to actual problem type (e.g., answerable or unanswerable) rather than dataset artifacts, we draw answerable and unanswerable items from similar embedding neighborhoods and explicitly control question length within

	Answer Unknown			False Premise			Underspecified Context			Overall		
Model	**Ans.**	**Abst.**	**TPC**	**Ans.**	**Abst.**	**TPC**	**Ans.**	**Abst.**	**TPC**	**Ans.**	**Abst.**	**TPC**
					Without Search							
GPT-4o-mini	41.8	65.8	157.3	54.7	67.4	105.9	76.1	27.2	264.9	57.5	53.5	176.0
o4-mini	46.6	65.1	820.2	57.8	65.3	722.3	83.2	26.6	623.3	62.5	52.3	721.9
Kimi-K2	49.0	63.0	255.8	58.3	63.2	101.6	79.2	23.8	306.3	62.2	50.0	221.2
Qwen3-235B-Instruct	47.2	64.8	268.2	55.7	69.3	180.0	79.3	24.2	395.2	60.7	52.8	281.1
Qwen3-235B-Think	50.0	64.4	1155.2	57.3	63.5	1039.1	79.4	31.9	1159.8	62.2	53.3	1118.0
Hermes3-3B	17.1	80.5	91.7	24.0	83.4	60.6	53.5	32.2	212.4	35.0	60.8	133.0
Llama-3.2-3B	27.4	57.5	255.6	41.1	77.7	146.6	61.3	25.4	320.8	43.3	53.5	241.0
Llama-3.3-70B	46.6	59.6	338.4	56.2	68.4	177.6	76.5	28.0	355.7	59.8	52.0	290.6
Mistral-Small-24B	40.4	64.6	257.5	52.1	67.9	173.0	75.8	29.7	327.5	56.1	54.1	252.7
Average	40.7	65.0	399.9	50.8	69.6	300.7	73.8	27.7	440.6	55.5	54.7	381.9
					With Search							
GPT-4o-mini	63.0	62.3	942.4	67.2	61.1	777.1	84.8	19.5	762.9	71.7	47.6	827.5
o4-mini	63.4	64.4	1031.8	68.8	60.0	1155.3	87.5	23.3	871.3	73.2	49.2	1019.5
Kimi-K2	64.4	61.6	851.8	67.7	65.8	565.9	85.5	24.2	553.0	72.5	50.5	656.9
Qwen3-235B-Instruct	64.4	66.9	923.0	66.7	68.2	652.1	85.2	22.3	859.5	72.1	52.5	811.5
Qwen3-235B-Think	63.7	64.8	1292.9	69.3	65.1	1245.1	85.5	23.7	1338.9	72.8	51.2	1292.3
Hermes3-3B	45.9	35.6	493.4	56.8	33.7	560.6	57.0	13.2	369.2	54.2	27.5	461.9
Llama-3.2-3B	58.2	61.6	717.8	60.9	64.2	681.3	73.4	21.5	804.7	64.2	49.1	734.6
Llama-3.3-70B	62.3	62.3	731.5	68.2	62.7	685.2	83.5	20.6	834.7	71.3	48.5	750.5
Mistral-Small-24B	56.8	64.1	329.2	62.5	65.3	246.5	83.2	30.1	414.0	67.5	53.2	329.9
Average	60.2	60.4	812.6	65.3	60.7	729.9	80.6	22.0	756.5	68.8	47.7	765.0

Table 2: Over-searching behavior across query types. Search augmentation consistently improves answer accuracy but degrades abstention accuracy, with Underspecified Context questions exhibiting the most severe degradation.

each category. Figure 3 demonstrates the effectiveness of our filtering process, showing similar length distributions and high semantic similarity between answerable and unanswerable questions across all categories. See Appendix D for full curation details and statistics.

Following Kirichenko et al. (2025), we create OVERSEARCHQA based on three categories: **Answer Unknown (AU)** – future events and unsolved problems; **False Premise (FP)** – incorrect assumptions or contradictory claims; and **Underspecified Context (UC)** – ambiguous intent or missing information requiring clarification. A concise category summary is shown in Table 1.

Models. We evaluate over-searching behavior across a diverse set of models, including both open-source and API-based: GPT-4o-mini (Hurst et al., 2024), Kimi-K2 (Kimi et al., 2025), Qwen3-235B-Instruct (Yang et al., 2025), Llama-3.2-3B (Grattafiori et al., 2024), Llama-3.3-70B (Grattafiori et al., 2024), Mistral-Small-24B (Mistral, 2025), o4-mini (OpenAI, 2025b), Qwen3-235B-Thinking (Yang et al., 2025), Hermes3-3B (Teknium et al., 2024), and o4-mini-deep-research (OpenAI, 2025a). Each model is evaluated both with and without search augmentation to isolate the impact of search on abstention behavior. The deep research system has search enabled by default, with results reported separately in Figure 4. For reasoning models (o4-mini and Qwen3-235B-Thinking), reasoning effort is set to default. To ensure fair comparison across all search-augmented models, we maintain identical retrieval infrastructure, such as top-k retrieved documents and retrievers. Unless otherwise noted, we use Wikipedia (enwiki-20250801) with E5-base (Wang et al., 2022) as the default retriever. Models are permitted up to 10 search calls per query. We compare different retrieval sources in §5.2. Complete setup details are provided in Appendix E.

5 Results

5.1 Search Augmentation Harms Abstention

Search Improves Answer Accuracy but Degrades Abstention. Table 2 shows that while incorporating search improves accuracy on answerable questions, it simultaneously impairs the models' ability to abstain from unanswerable ones, boosting answer accuracy by an average of 24.0% while degrading abstention accuracy by 12.8%. This negative effect is most pronounced on Underspecified Context questions, where models attempt to find supporting evidence for queries that are fundamentally unanswerable. Conversely, models

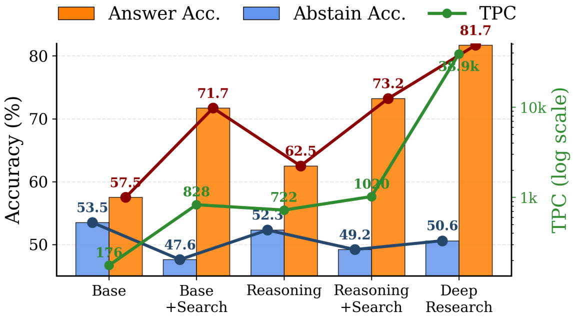

Figure 4: Comparison of the same model family with different configurations: Base (`GPT-4o-mini`), Reason (`o4-mini`), and Deep Research (`o4-mini-deep-research`). Answer accuracy increases while abstention accuracy consistently degrades as configurations become more complex. TPC (shown in log scale) increases with search capabilities; Deep Research dramatically reaches 38.9k TPC – over 221× compared to the base configuration.

Metric	Low	Medium	High
Ans. Acc	74.1	74.3	74.6
Abst. Acc	46.6	46.2	45.4
Overall Acc	60.4	60.3	60.0
TPC	517.1	1002.7	1492.2

Table 3: Impact of different reasoning effort levels on o4-mini. Answer accuracy increases with more reasoning effort, but abstention accuracy decreases. TPC increases monotonically with reasoning effort.

achieve higher answer accuracy when the missing context is explicitly provided for these same questions. Detailed case studies for three categories can be found in Appendix H.

Reasoning and Model Complexity Amplify Over-Searching. To understand the impact of reasoning and model complexity, we analyze different levels of reasoning effort on o4-mini. Table 3 shows that while more reasoning consistently improves answer accuracy, it degrades abstention accuracy. TPC increases monotonically with reasoning effort, suggesting that deeper reasoning may encourage models to over-search. Additionally, Figure 4 illustrates this trade-off within the same model family across different model complexity: adding search capabilities consistently improves answer accuracy at the cost of abstention. The Deep Research configuration, for example, reaches the highest answer accuracy but requires significant computational resources, suggesting that increased complexity amplifies over-searching.

Abstention Failure Costs the Most. We further analyze TPC by decomposing it across outcome categories. Figure 5 shows that *abstention failure* (i.e., answering unanswerable queries) remains the highest TPC for most models, where models repeatedly invoke search for fundamentally unanswerable queries, accumulating larger costs without achieving correctness.

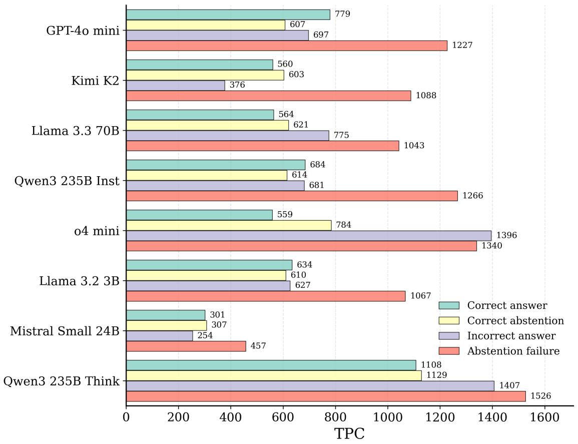

Figure 5: TPC breakdown by outcome categories. Abstention failure remains the most expensive behavior for most models.

5.2 Retrieval Matters

Noisy Retrieval Causes More Search. We compare four retrieval sources to understand how corpus quality affects over-searching: (i) *Wikipedia-Latest*, the most reliable source with up-to-date documents (from 2025); (ii) *Wikipedia-Stale*, using an outdated Wikipedia snapshot (from 2018); (iii) *C5*, a noisy corpus from Vanroy (2025) with Wikipedia content removed; and (iv) *Web Search*, real-world online search. More details on retrieval setup are provided in Appendix E.2.

Table 4 shows that corpus quality has a significant impact on over-searching. C5 exhibits dramatically higher TPC (3.6× on average) than Wikipedia-Latest, indicating that models perform much more searches when retrieval quality is poor. Interestingly, C5 also achieves the second-best abstention accuracy, suggesting that consistently poor retrieval may paradoxically help models recognize unanswerability. Web Search achieves the best answer accuracy but lower abstention accuracy. This may be because of its access to the full internet, where search results may directly contain answers to questions, while the abundance of mixed signals from diverse web sources makes it difficult for models to recognize when a question is unanswerable. This reflects the challenges of real-world retrieval environments where uncontrollable and mixed signals can complicate abstention decisions.

	Wikipedia-Latest			**Wikipedia-Stale**			**C5**			**Web Search**		
Model	**Ans.**	**Abst.**	**TPC**	**Ans.**	**Abst.**	**TPC**	**Ans.**	**Abst.**	**TPC**	**Ans.**	**Abst.**	**TPC**
GPT-4o-mini	71.7	47.6	827.5	71.0	46.2	1124.1	69.3	48.8	2350.6	71.0	47.6	645.2
o4-mini	73.2	49.2	1019.5	72.7	46.7	1170.3	72.4	48.2	3311.7	74.4	47.0	1239.3
Kimi-K2	72.5	50.5	656.9	71.9	49.1	904.2	71.7	50.9	3147.9	73.2	45.8	741.3
Qwen3-235B-Instruct	72.1	52.5	811.5	72.9	49.7	997.4	71.2	51.8	3794.1	74.1	47.0	1165.4
Mistral-Small-24B	67.5	53.2	329.2	66.9	50.2	428.9	65.4	51.7	1486.9	68.4	47.5	684.1
Llama-3.3-70B	71.3	48.5	750.5	71.2	45.7	776.8	70.5	48.9	1548.8	72.7	44.1	936.9
Average	71.4	50.2	732.5	71.1	47.9	900.3	70.1	50.1	2606.7	72.3	46.5	902.0

Table 4: Impact of retrieval quality on over-searching behavior. Noisy retrieval (C5) causes models to perform additional searches, dramatically increasing TPC.

	GPT-4o-mini		**o4-mini**		**Qwen3-Inst**		**Kimi-K2**		**Llama3.3-70B**		**Mistral-Small-24B**	
Evid.	**Acc.**	**Evid.**	**Acc.**	**Evid.**	**Acc.**	**Evid.**	**Acc.**	**Evid.**	**Acc.**	**Evid.**	**Acc.**	**Evid.**
Only Positive	18.0	0.0	16.3	0.0	17.4	0.0	19.6	0.0	17.0	0.0	16.2	0.0
Pos≥Neg	56.7	32.5	57.1	32.9	41.3	32.8	36.0	31.2	55.9	33.3	54.9	33.1
Neg>Pos	73.8	67.5	74.4	67.1	83.9	68.2	72.4	68.8	77.6	66.7	75.1	66.9
Only Negative	91.1	100.0	89.4	100.0	98.6	100.0	92.9	100.0	92.6	100.0	89.7	100.0

Table 5: Abstention accuracy on unanswerable queries grouped by naturally retrieved evidence balance. Rows represent queries categorized by the balance of positive vs. negative evidence naturally retrieved during inference. "Evid." columns show the percentage of queries in each category. Models achieve near-perfect abstention with only negative evidence, but degrade sharply when positive evidence dominates.

Abstention Cues Are Rare. Real-world corpora overwhelmingly document what we know, not what we don't know. This asymmetry could create a bias where models interpret unknowability as inadequate search effort. We investigate the nature of retrieved documents and whether such bias impacts abstention. We leverage an LLM judge to classify naturally retrieved documents into: *positive documents* containing answer-supporting evidence (for unanswerable queries, this means misleading information), and *negative documents* indicating unanswerability (e.g., uncertainty statements, contradictions). We group unanswerable queries by their naturally retrieved evidence balance. Table 5 shows models achieve near-perfect abstention when only negative evidence is present, but degrade sharply when positive evidence dominates. However, negative documents comprise only 13-22% of retrieved content for unanswerable queries (Table 10), contributing to the lack of abstention behavior. Details of the classification procedure can be found in Appendix F.1.

5.3 Snowball in Multi-turn Conversations

We investigate how multi-turn conversational settings impact models' abstention abilities. We construct conversations of 1–9 turns, where the final-turn query remains fixed for evaluation. We evaluate three conversational contexts: (i) *Unanswerable*, where all preceding turns contain unanswerable questions; (ii) *Mixed*, with a random mix of answerable and unanswerable questions; and (iii) *Answerable*, where all preceding turns contain answerable questions. Figure 6 shows the results for GPT-4o-mini. For the unanswerable context, abstention accuracy remains relatively stable with even slight improvement as conversation turns increase, suggesting that repeated exposure to unanswerable queries and potential abstention helps models maintain abstention patterns. In contrast, answerable and mixed contexts exhibit degradation in abstention, suggesting that prior answerable questions bias the model toward attempting answers. Meanwhile, TPC increases with conversation length for all contexts. These findings reveal a snowball effect where models carry forward accumulated search patterns from earlier turns – a history of unanswerable questions encourages abstention, while a history of answerable questions encourages answer attempts.

5.4 Mitigating Over-Searching

We explore two training-free strategies for over-searching mitigation: *query-level mitigation*, which improves system prompt and workflow design, and *retrieval-level mitigation*, which augments the corpus with negative evidence to facilitate abstention.

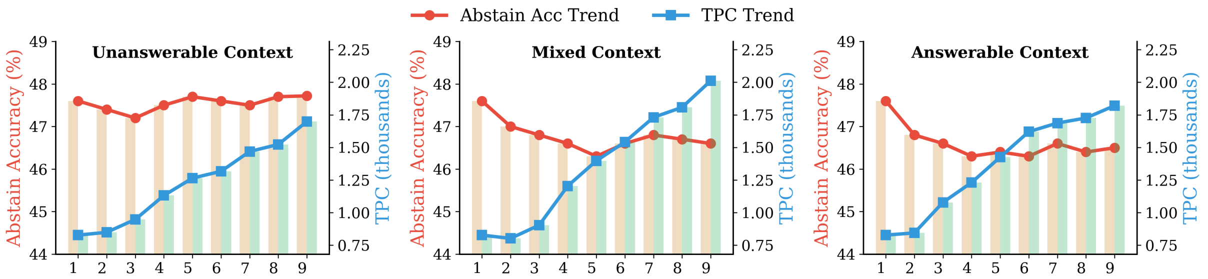

Figure 6: Multi-turn conversations amplify over-searching behavior. Unanswerable context maintains stable abstention accuracy and even shows slight improvement across turns, while Answerable context exhibits the largest abstention degradation. TPC increases with conversation length for all contexts.

	Baseline			Abstention-aware			Few-shot			Self-eval			Corpus Aug.		
Model	**Ans.**	**Abst.**	**TPC**	**Ans.**	**Abst.**	**TPC**	**Ans.**	**Abst.**	**TPC**	**Ans.**	**Abst.**	**TPC**	**Ans.**	**Abst.**	**TPC**
GPT-4o-mini	71.7	47.6	827.5	69.7	53.2	346.8	67.5	67.1	270.0	65.6	63.1	545.8	71.2	50.7	843.6
o4-mini	73.2	49.2	1019.5	72.7	52.5	852.8	72.2	59.8	792.5	71.9	57.4	973.9	72.8	53.0	962.3
Kimi-K2	72.5	50.5	656.9	71.9	62.3	474.4	72.2	67.5	542.3	72.4	62.5	656.8	71.9	54.7	665.2
Qwen3-235B-Instruct	72.1	52.5	811.5	72.6	68.8	677.4	72.1	59.9	853.5	70.4	61.8	774.5	71.6	56.6	823.1
Mistral-Small-24B	67.5	53.2	329.9	66.8	58.4	285.3	66.2	60.8	312.7	67.1	60.2	318.5	67.3	55.9	341.2
Llama-3.3-70B	71.3	48.5	750.5	65.8	65.7	691.1	67.5	65.0	730.3	71.9	63.9	713.9	70.8	52.1	782.9
Average	71.4	50.2	732.6	69.9	60.2	554.8	69.6	63.4	583.6	69.9	61.5	663.9	70.9	53.8	736.4

Table 6: Evaluation of mitigation strategies for over-searching. Query-level approaches (Abstention-aware, Few-shot, Self-eval) modify system prompts, while the retrieval-level approach (Corpus Aug.) augments the corpus with synthetic negative evidence documents.

Query-Level Mitigation. We evaluate three prompt-based methods: **(1) Abstention-aware** explicitly instruct models to consider abstention as a valid response when queries are unanswerable; **(2) Few-shot learning** provides examples of appropriate abstention behavior in the system prompt; and **(3) Self-evaluation** introduces a self-assessment stage where the model evaluates query answerability before answering. Table 6 shows that all three methods substantially improve abstention accuracy, achieving an average gain of 11.5 percentage points. Few-shot learning achieves the strongest abstention improvements but incurs the largest answer accuracy reduction, suggesting that explicit examples may bias models toward over-abstention. Self-evaluation achieves balanced improvements in abstention with modest answer accuracy loss, though it exhibits higher TPC due to additional reasoning and potential searches required for self-assessment. While query-level interventions could reduce over-searching, they introduce different trade-offs between answer accuracy, abstention behavior, and computational cost. Prompt templates for all strategies are provided in Appendix G.

Retrieval-Level Mitigation. Table 5 shows that negative evidence improves abstention when present. Therefore, we evaluate **corpus augmentation** for over-searching mitigation by inserting 10 synthetic negative evidence for all queries into the corpus (see Appendix F.2 for more details). Table 6 shows modest improvements (3.6% on average) in abstention accuracy. This limited effectiveness may occur because: (i) synthetic documents rank poorly in retrieval; (ii) negative evidence is diluted by numerous naturally-occurring positive documents. While negative evidence helps when retrieved, effective retrieval-level mitigation would require systematic architectural changes, which we leave for future work.

6 Conclusion

In this work, we conduct a comprehensive evaluation and demonstrate the “over-search” behavior in search-augmented LLMs, where search tools are invoked unnecessarily, leading to increased computational costs and potential degradation in response quality. Our systematic evaluation reveals a fundamental trade-off: while search improves accuracy on answerable queries, it impairs the model’s ability to abstain from unanswerable ones. This phenomenon is particularly pronounced in reasoning models, complex systems, with noisy retrieval, and in multi-turn conversations where search behavior can snowball. We introduce the Tokens Per Correctness (TPC) metric to quantify this inefficiency and show that negative evidence in search results significantly improves abstention. We evaluate query-level and retrieval-level mitigation strategies and find that while both can help mitigate over-searching to some extent, they do not resolve models' fundamental inability to search rationally. Finally, we release OVERSEARCHQA to foster continued research into improving search efficiency and abstention capabilities in tool-augmented LLMs.

7 Limitations

In this work, we focus on comprehensively evaluating and analyzing over-searching behavior. We investigate several training-free mitigation strategies; however, other promising directions remain, including targeted model training and architectural modifications to the retrieval system. We leave these aspects for future exploration. Furthermore, our unanswerable queries in OVERSEARCHQA are curated from existing benchmarks rather than collected from real-world search logs. While this allows us to isolate the model's decision-making failures from confounding factors like retrieval failure, it may not reflect the distribution of unanswerable queries in deployment and can be outdated. Real-world user queries may exhibit different linguistic patterns or types of unanswerability that are not fully captured by our categories. Finally, while we evaluate query-level and retrieval-level mitigations, we find they offer only modest improvements, suggesting that addressing the inability of models to search rationally may require interventions at the post-training or alignment stage.

8 Acknowledgment

We thank Jiabao Ji, Priyanshu Kumar, Bowen Jin, Jinhao Lei, Minsoo Kim, Feijie Wu, and Donghan Yu for thoughtful discussions. Roy Xie is supported by the Apple Scholars in AI/ML PhD fellowship and NSF Graduate Research Fellowship.

References

Alfonso Amayuelas, Kyle Wong, Liangming Pan, Wenhu Chen, and William Yang Wang. 2024. Knowledge of knowledge: Exploring known-unknowns uncertainty with large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6416–6432, Bangkok, Thailand. Association for Computational Linguistics.

Youssef Benchekroun, Megi Dervishi, Mark Ibrahim, Jean-Baptiste Gaya, Xavier Martinet, Grégoire Mialon, Thomas Scialom, Emmanuel Dupoux, Dieuwke Hupkes, and Pascal Vincent. 2023. Worldsense: A synthetic benchmark for grounded reasoning in large language models. *arXiv preprint arXiv:2311.15930*.

Faeze Brahman, Sachin Kumar, Vidhisha Balachandran, Pradeep Dasigi, Valentina Pyatkin, Abhilasha Ravichander, Sarah Wiegreffe, Nouha Dziri, Khyathi Chandu, Jack Hessel, and 1 others. 2024. The art of saying no: Contextual noncompliance in language models. *Advances in Neural Information Processing Systems*, 37:49706–49748.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Fan Yang, Zenan Zhou, Weipeng Chen, Haofen Wang, Jeff Z Pan, and 1 others. 2025. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Trans. Mach. Learn. Res.*, 2023.

Mingyi Deng, Lijun Huang, Yani Fan, Jiayi Zhang, Fashen Ren, Jinyi Bai, Fuzhen Yang, Dayi Miao, Zhaoyang Yu, Yifan Wu, and 1 others. 2025. Interactcomp: Evaluating search agents with ambiguous queries. *arXiv preprint arXiv:2510.24668*.

Mehmet Hamza Erol, Batu El, Mirac Suzgun, Mert Yuksekgonul, and James Zou. 2025. Cost-of-pass: An economic framework for evaluating language models. *arXiv preprint arXiv:2504.13359*.

Chenrui Fan, Ming Li, Lichao Sun, and Tianyi Zhou. 2025. Missing premise exacerbates overthinking: Are reasoning models losing critical thinking skill? *arXiv preprint arXiv:2504.06514*.

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024. Don't hallucinate, abstain: Identifying LLM knowledge gaps via multi-LLM collaboration. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14664–14690, Bangkok, Thailand. Association for Computational Linguistics.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. *ArXiv*, abs/2211.10435.

Google. 2024. Gemini deep research.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. 2025. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*.

Shengding Hu, Yifan Luo, Huadong Wang, Xingyi Cheng, Zhiyuan Liu, and Maosong Sun. 2023. Won't get fooled again: Answering questions with false premises. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5626–5643, Toronto, Canada. Association for Computational Linguistics.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Jiabao Ji, Min Li, Priyanshu Kumar, Shiyu Chang, and Saloni Potdar. 2025. Deepambigqa: Ambiguous multi-hop questions for benchmarking llm answer completeness. *Preprint*, arXiv:2511.01323.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025a. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, and Ji-Rong Wen. 2025b. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. In *Companion Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025 - 2 May 2025*, pages 737–740. ACM.

Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. 2025. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*.

Najoung Kim, Phu Mon Htut, Samuel R. Bowman, and Jackson Petty. 2023. $(\text{QA})^2$: Question answering with questionable assumptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8466–8487, Toronto, Canada. Association for Computational Linguistics.

Kimi, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, and 150 others. 2025. Kimi k2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J Bell. 2025. Abstentionbench: Reasoning llms fail on unanswerable questions. *arXiv preprint arXiv:2506.09038*.

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. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

LangGraph. 2025. Langgraph.

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.

Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan Ilgen, Emma Pierson, Pang Wei W Koh, and Yulia Tsvetkov. 2024. Mediq: Question-asking llms and a benchmark for reliable interactive clinical reasoning. *Advances in Neural Information Processing Systems*, 37:28858–28888.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. 2025a. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025b. Torl: Scaling tool-integrated rl. *arXiv preprint arXiv:2503.23383*.

Nishanth Madhusudhan, Sathwik Tejaswi Madhusudhan, Vikas Yadav, and Masoud Hashemi. 2024. Do llms know when to not answer? investigating abstention abilities of large language models. *arXiv preprint arXiv:2407.16221*.

Mistral. 2025. Mistral small 3.

OpenAI. 2025a. Deep research system card.

OpenAI. 2025b. Openai o3 and o4-mini system card.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman. 2022. BBQ: A hand-built bias benchmark for question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2086–2105, Dublin, Ireland. Association for Computational Linguistics.

Xiao Pu, Michael Saxon, Wenyue Hua, and William Yang Wang. 2025. Thoughtterminator: Benchmarking, calibrating, and mitigating overthinking in reasoning models. *arXiv preprint arXiv:2504.13367*.

Joykirat Singh, Raghav Magazine, Yash Pandya, and Akshay Nambi. 2025. Agentic reasoning and tool integration for llms via reinforcement learning. *arXiv preprint arXiv:2505.01441*.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025a. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*.

Linxin Song, Taiwei Shi, and Jieyu Zhao. 2025b. The hallucination tax of reinforcement finetuning. *arXiv preprint arXiv:2505.13988*.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and 1 others. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Ryan Teknium, Jeffrey Quesnelle, and Chen Guang. 2024. Hermes 3 technical report. *Preprint*, arXiv:2408.11857.

Christian Tomani, Kamalika Chaudhuri, Ivan Evtimov, Daniel Cremers, and Mark Ibrahim. 2024. Uncertainty-based abstention in llms improves safety and reduces hallucinations. *arXiv preprint arXiv:2404.10960*.

Bram Vanroy. 2025. Commoncrawl creativecommons corpus (c5).

Hongru Wang, Cheng Qian, Wanjun Zhong, Xiusi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong, and Heng Ji. 2025. Acting less is reasoning more! teaching model to act efficiently. *arXiv preprint arXiv:2504.14870*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*.

Bingbing Wen, Bill Howe, and Lucy Lu Wang. 2024. Characterizing LLM abstention behavior in science QA with context perturbations. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3437–3450, Miami, Florida, USA. Association for Computational Linguistics.

Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe, and Lucy Lu Wang. 2025. Know your limits: A survey of abstention in large language models. *Transactions of the Association for Computational Linguistics*, 13:529–556.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380. Association for Computational Linguistics.

Xunjian Yin, Baizhou Huang, and Xiaojun Wan. 2023a. ALCUNA: Large language models meet new knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1397–1414, Singapore. Association for Computational Linguistics.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023b. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.

Appendix

A Over-searching Definition & Discussion . . . 13
  A.1 Subjectivity of Over-Searching Thresholds . . . 13
  A.2 Over-Searching as Marginal Return on Investment . . . 13
  A.3 Empirical Demonstration of Over-Searching . . . 14
  A.4 Measuring Over-Searching with TPC . . . 15
B Additional Metric Details . . . 15
  B.1 Dual Accuracy . . . 15
  B.2 TPC Parameter Selection . . . 15
  B.3 TPC vs. Other Metrics . . . 15
C LLM as Judge . . . 16
  C.1 Inter-Judge Agreement . . . 17
  C.2 Human Annotation Validation . . . 17
  C.3 Dual Accuracy Evaluation Prompt . . . 18
D Dataset Details . . . 20
E Setup Details . . . 21
  E.1 Model Details . . . 21
  E.2 Retrieval Setup . . . 21
F Abstention Cues Analysis . . . 21
  F.1 Classification of Naturally Retrieved Documents . . . 21
  F.2 Generation of Synthetic Negative Evidence . . . 22
G Mitigation Strategy Prompts . . . 23
H Case Studies . . . 24

A Over-searching Definition & Discussion

We define over-searching as occurring when additional search operations yield *disproportionate cost relative to correctness gains*. An increasing TPC indicates over-searching because it indicates each additional correct response requires more computational resources. Importantly, even if absolute accuracy increases, over-searching can still occur if the token cost grows at a faster rate than the accuracy improvement, resulting in diminishing returns. In this section, we discuss in detail on the nuance about over-searching definition.

A.1 Subjectivity of Over-Searching Thresholds

Over-searching is an inherently subjective concept, as the point at which additional search effort becomes inefficient depends on the goals, constraints, and priorities of the application. In casual conversational contexts, efficiency and responsiveness are typically prioritized, making an early stopping point adequate. In business intelligence tasks, the objective often lies in balancing accuracy with computational cost, resulting in a moderate search depth. In medical diagnosis, accuracy holds higher importance, and extended searching can be justified if it reduces the probability of error. Legal research, on the other hand, demands exhaustive coverage, where the notion of over-searching becomes less meaningful. Hence, defining universal thresholds for over-searching is impractical, as it exists on a continuum rather than a binary condition.

A.2 Over-Searching as Marginal Return on Investment

A quantifiable way to measure over-searching is to compute the marginal return on investment (ROI) for each additional search:

$$\text{ROI}_j = \frac{\Delta\text{Accuracy}_{j-1\to j}}{\Delta\text{Cost}_{j-1\to j}/k} \times 100\% \tag{2}$$

where $\Delta\text{Accuracy}_{j-1\to j}$ represents the accuracy improvement from search number $j-1$ to j, and $\Delta\text{Cost}_{j-1\to j}$ is the marginal token cost (normalized per k tokens). Intuitively, ROI computes the accuracy gain per k tokens spent. We set $k = 1000$ and compute ROI for the same model from Figure 2. Table 7 shows the contrast between initial search value and subsequent searches. The first search provides exceptional ROI (0.874%), improving overall accuracy (average between abstention accuracy and answer accuracy) by 2.45% from the no-search baseline. However, ROI decreases dramatically for additional searches, with Max 5 onwards yielding negative or near-zero ROI. Notably, turns 15 and 17 show strong negative ROI, indicating pure over-searching where computational costs yield no accuracy benefit and in fact coincide with accuracy decline.

Max Turn	Accuracy (%)	TPC	Marginal Gain (%)	Marginal Cost	ROI
0	57.40	721.9	–	–	–
1	59.85	3,526.6	+2.45	2,804.7	+0.874
3	60.85	4,660.6	+1.00	1,134.0	+0.882
5	60.80	6,190.0	−0.05	1,529.3	−0.033
7	60.70	6,509.8	−0.10	319.8	−0.313
9	60.75	7,490.2	+0.05	980.4	+0.051
11	60.80	7,665.6	+0.05	175.5	+0.285
13	60.70	8,437.7	−0.10	772.1	−0.130
15	60.25	8,719.9	−0.45	282.1	−1.595
17	59.95	8,802.4	−0.30	82.5	−3.634
19	60.05	9,120.4	+0.10	318.0	+0.314

Table 7: Marginal ROI analysis across search turns. The dramatic ROI drop from the first search to subsequent searches demonstrates severe diminishing returns.

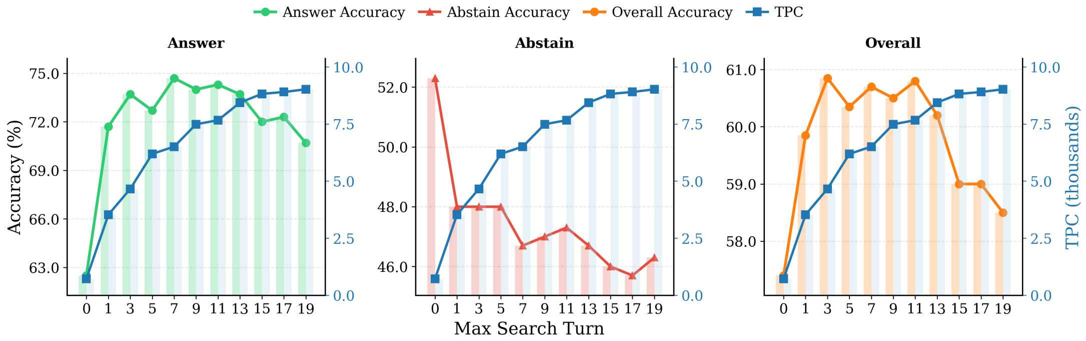

Figure 7: Detailed breakdown of performance vs. maximum search turns (extended view of Figure 2). This shows how answer accuracy (blue circles, measured on answerable queries), abstention accuracy (orange triangles, measured on unanswerable queries), and Tokens Per Correctness (green squares) evolve for o4-mini as the maximum number of search calls increases from 0 to 19. Answer accuracy initially improves, reaches a peak around 7 searches, and then declines with excessive searching, while TPC continues rising from 722 to over 9k tokens per correct response. Critically, abstention accuracy degrades from 52.3% to 46.3%, demonstrating that additional search calls actively harm the model's ability to recognize unanswerable queries.

Model	Actual Search	Optimal Search	Over-Search (%)
GPT-4o-mini	0.826	0.471	75.4
o4-mini	0.414	0.253	63.6
Kimi-K2	0.455	0.256	77.7
Qwen3-235B-Instruct	0.830	0.488	70.1
Mistral-Small-24B	0.236	0.165	43.0
Llama-3.3-70B	0.958	0.518	84.9
Average	*0.620*	*0.364*	*70.5*

Table 8: Measurement of over-searching. On average, models perform 0.620 searches when only 0.364 are needed, corresponding to a 70.5% over-search rate.

A.3 Empirical Demonstration of Over-Searching

In §3.1, we defined over-searching at the aggregate level due to the noise inherent in individual model trajectories. However, as a concrete empirical demonstration, we analyze over-searching at the instance level by assuming that the first time a model reaches a correct state is its "optimal" stopping point.

We compare a model's actual number of searches against the minimum required. First, we identify the subset of queries $\mathcal{D}_{\text{correct}} \subset \mathcal{D}$ where a model produces a correct response (a correct answer for $q \in \mathcal{A}$ or a correct abstention for $q \in \mathcal{U}$). Let the sequence of searches performed for such a query q be S_q, with a total of $k_q = |S_q|$ search calls.

Similar to Wang et al. (2025), to find the *optimal number of searches* k_q^*, we evaluate the model's response by truncating the search sequence. We force the model to predict using only the first t searches, $S_{1:t}$, for $t = k_q, k_q - 1, \ldots, 0$. The optimal number k_q^* is the minimum number of searches required to achieve the same correct outcome: $k_q^* = \min\{t \geq 0 : A(q, S_{1:t}) = 1\}$.

The *actual number of searches* k_q represents the model's natural behavior. Over-searching at the query level is the number of excess searches, $k_q - k_q^*$. Table 8 presents the average actual searches ($\bar{k}_q$) and average optimal searches ($\bar{k}_q^*$) across all queries in $\mathcal{D}_{\text{correct}}$. The Over-Search (%) column quantifies the average excess search relative to the optimal, calculated as $(\bar{k}_q/\bar{k}_q^*) - 1$. The results show that models perform 70.5% more searches on average than are necessary to achieve correctness.

While this analysis provides a concrete measure of over-searching, this "optimal search" calculation is not used as our primary evaluation metric due to several reasons. Firstly, this method is a computationally expensive approximation for large-scale evaluations, as it requires multiple model inferences for each query. Secondly, its scope is limited: it only applies to the subset of queries that the model already answered correctly ($\mathcal{D}_{\text{correct}}$). It fails to account for the inefficient costs accumulated on queries where the model's final response was incorrect. Therefore, we use TPC as our primary, intuitive alternative, as it captures the cost-performance trade-off across the *entire* dataset, where reducing TPC directly implies a reduction in over-searching.

A.4 Measuring Over-Searching with TPC

In this work, we focus on measuring relative rather than absolute over-searching through TPC. By comparing the same model with and without search augmentation, we isolate the specific contribution of search behavior while keeping all other factors constant. A higher TPC in the search-augmented case indicates *inefficient utilization of search relative to performance gain*. TPC naturally integrates both answerable and unanswerable queries through the $\mathrm{Correct}(q)$ function in Equation 1: for answerable queries, $\mathrm{Correct}(q) = 1$ if the model answers correctly; for unanswerable queries, $\mathrm{Correct}(q) = 1$ if the model appropriately abstains. Thus, TPC captures the full cost-effectiveness of search across both query types, penalizing systems that accumulate search costs without improving either answer accuracy (on $\mathcal{A}$) or abstention accuracy (on $\mathcal{U}$). A model that searches excessively on unanswerable queries without learning to abstain will incur high costs with low correctness, yielding increased TPC.

When TPC increases, it reflects that extra tokens are being spent without proportional correctness gains, which directly indicates over-searching. Conversely, reducing TPC implies reducing over-searching, as it means achieving correctness with fewer tokens and fewer searches. This relationship holds for both answerable and unanswerable queries: for answerable queries, over-searching manifests as unnecessary searches beyond the point where correctness is achieved; for unanswerable queries, over-searching occurs when models continue searching despite already having sufficient information to abstain appropriately.

B Additional Metric Details

We discuss in detail our evaluation metrics and explain the rationale for using them and the parameter choices for the TPC calculation.

B.1 Dual Accuracy

Prior work on abstention typically reports *abstention recall* (Kirichenko et al., 2025), which is the proportion of responses where the model correctly abstained. Let $\mathcal{U}$ denote unanswerable queries and $\mathcal{A}$ denote answerable queries. *Abstention recall* is defined as the fraction of $q \in \mathcal{U}$ where the model abstains (correct abstentions). Our *Dual Accuracy* aligns with and extends these metrics in a more intuitive way. Dual Accuracy separately reports **answer accuracy on $\mathcal{A}$** (what fraction of *answerable* queries are answered correctly?) and **abstention accuracy on $\mathcal{U}$** (what fraction of *unanswerable* queries are correctly abstained from? This is *identical* to abstention recall). By explicitly reporting two accuracies, we provide a symmetric, interpretable view of model behavior across both query types. This disentangles the two decision regimes – making it immediately clear how well a model answers when it should answer, and abstains when it should abstain – without needing to reason about precision, class imbalance, or aggregated metrics. The dual framing makes performance transparent and comparable across different datasets and task compositions.

B.2 TPC Parameter Selection

To ensure a standardized measure of over-searching, we normalize all costs using a single reference pricing based on popular production models. We use constant fixed values ($\lambda = 0.25, \mu = 500$), where λ represents the ratio of input-to-output token cost and μ represents the cost of a single search API call in equivalent output tokens. These values are derived from the public pricing of models like GPT-4o-mini (0.15 per 1M input, 0.60 per 1M output, giving λ = 0.25) and a standard search API (5 per 1000 queries). At a cost of 0.0006 per output token, one search equates to approximately 500 tokens. While individual model pricing varies, this approach provides a consistent evaluation of search efficiency between different models.

B.3 TPC vs. Other Metrics

TPC vs. Marginal ROI. While both TPC and marginal ROI provide insights into search efficiency, they serve complementary roles in our analysis. Marginal ROI measures the *per-turn* efficiency by computing accuracy gains relative to incremental token costs. In contrast, TPC provides a *cumulative* measure that aggregates total token expenditure per correct answer across an entire dataset. The two metrics are consistent in their implications. Table 7 demonstrates that marginal ROI is strongly positive for the initial search turns (0.874 and 0.882 for Max 1 and Max 3 respectively), indicating that early searches provide substantial value. Beyond Max 3, marginal ROI becomes erratic and frequently negative, suggesting that additional search turns provide negligible or even detrimental returns. This aligns perfectly with the TPC trends in Figure 2, where TPC increases monotonically after Max 3 while overall accuracy plateaus, confirming diminishing returns.

We use TPC as our primary metric in the main evaluation because TPC provides a single aggregate measure that is robust and interpretable, naturally handling cases where marginal accuracy changes are zero or negative (as seen in Max 5, 7, 13, 15, and 17), which would require careful interpretation under ROI. Additionally, TPC enables direct comparison across models with different search behaviors without requiring alignment of search turn boundaries. While marginal ROI is valuable for understanding *where* diminishing returns begin, TPC efficiently quantifies the *overall* efficiency of search-augmented systems – which is the central focus of this study.

TPC vs. CoP. Recent work proposes Cost-of-Pass (CoP) to quantify accuracy–cost trade-offs (Erol et al., 2025). While compelling for general benchmarking, TPC is a better fit for *tool-use efficiency* for several reasons. First, TPC provides *tool-aware costing* by decomposing compute into generated tokens, input/context tokens, and explicit tool/search calls, reflecting real costs in tool-augmented systems. Second, TPC offers *dataset-level stability* by aggregating as total cost over total correct across the dataset, avoiding pathologies from per-problem infinities when items are never solved. Third, TPC enables *apples-to-apples comparison* by fixing coefficients for input tokens and per-search actions, allowing direct comparison of tool vs. non-tool models under a standardized cost model. We therefore report TPC in the main text for its robustness and direct relevance to the costs incurred by search-augmented models.

C LLM as Judge

Our large language model judge system uses GPT-4o-mini as the default judge. We employ a modified version of the judge prompt adapted from Kirichenko et al. (2025) for abstention accuracy, which demonstrated high correlation between human annotation and judge evaluation using a Llama-8B model. For answer accuracy, we adapt the prompt directly from Wei et al. (2024), which similarly demonstrated strong agreement with human annotation.

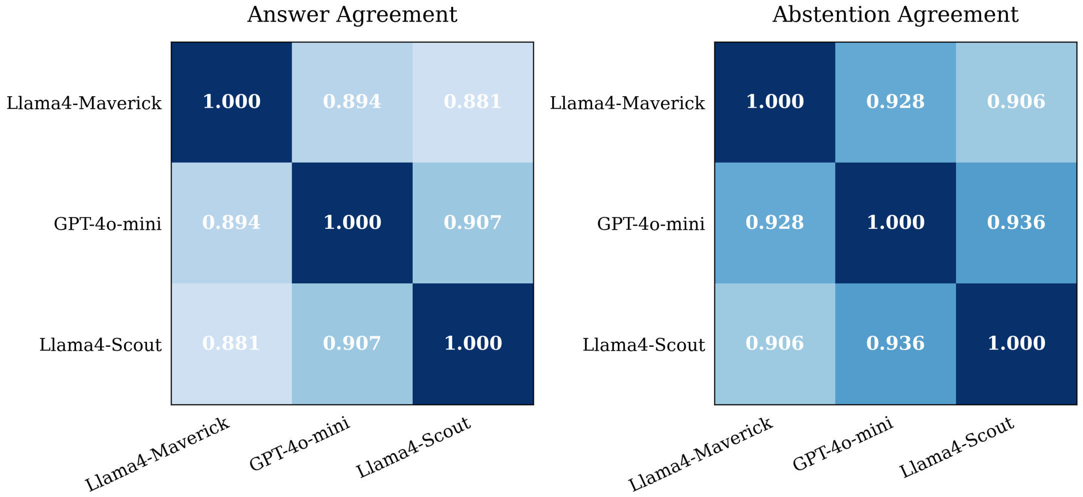

Figure 8: Pairwise agreement matrix between three independent LLM judges for answer accuracy (left) and abstention accuracy (right). High agreement scores demonstrate the reliability and consistency of the LLM-as-Judge evaluation framework across different model judges.

C.1 Inter-Judge Agreement

To ensure the robustness and reliability of our evaluation judge, we first evaluate consistency across three independent LLM judges for responses from GPT-4o-mini: Llama-4-Scout, Llama-4-Maverick, and GPT-4o-mini. Figure 8 shows the pairwise agreement between these judges for both answer accuracy and abstention accuracy. For answer accuracy, we observe an average pairwise agreement of 89.4% across all judge pairs. For abstention accuracy, the inter-judge consistency is even higher, with an average pairwise agreement of 92.3%. The high inter-judge consistency, particularly for abstention accuracy, confirms that our evaluation framework produces reliable and reproducible results across different model judges.

C.2 Human Annotation Validation

Beyond inter-judge agreement, we validate the reliability of our LLM judge against human expert judgment. We randomly selected 100 GPT-4o-mini responses from unanswerable queries and had them annotated. A human annotator evaluated these responses for abstention accuracy using the same criteria as the LLM judge. We compare the agreement between human judgment and the default LLM judge (GPT-4o-mini). The overall agreement rate reaches 84%. Analysis of the 16 disagreement cases (representing a disagreement rate of 16%) shows that the 10/16 disagreements occurred in one direction: the LLM judge identified abstention in cases where the human annotator did not. This conservative bias is acceptable for our evaluation purposes, as it does not systematically favor any particular model.

C.3 Dual Accuracy Evaluation Prompt

Answer Accuracy Evaluation

Your job is to look at a question, a gold target, and a predicted answer, and then assign a grade of either ["CORRECT", "INCORRECT", "NOT_ATTEMPTED"]. First, I will give examples of each grade, and then you will grade a new example.
The following are examples of CORRECT predicted answers.

```
Question: What are the names of Barack Obama's children?
Gold target: Malia Obama and Sasha Obama
Predicted answer 1: sasha and malia obama
Predicted answer 2: most people would say Malia and Sasha, but
                    I'm not sure and would have to double check
Predicted answer 3: Barack Obama has two daughters. Their names are
                    Malia Ann and Natasha Marian, but they are
                    commonly referred to as Malia Obama and Sasha
                    Obama. Malia was born on July 4, 1998, and
                    Sasha was born on June 10, 2001.
```

These predicted answers are all CORRECT because:

- They fully contain the important information in the gold target.
- They do not contain any information that contradicts the gold target.
- Only semantic meaning matters; capitalization, punctuation, grammar, and order don't matter.
- Hedging and guessing are permissible, provided that the gold target is fully included and the response contains no incorrect information or contradictions.

The following are examples of INCORRECT predicted answers.

```
Question: What are the names of Barack Obama's children?
Gold target: Malia and Sasha
Predicted answer 1: Malia.
Predicted answer 2: Malia, Sasha, and Susan.
Predicted answer 3: Barack Obama does not have any children.
Predicted answer 4: I think it's either Malia and Sasha. Or it
                    could be Malia and Jackie. Or it could be
                    Joey and Malia.
```

These predicted answers are all INCORRECT because:

- A factual statement in the answer contradicts the gold target. Incorrect statements that have some hedging (e.g., "it is possible that", "although i'm not sure, i think") are also considered incorrect.

The following are examples of NOT_ATTEMPTED predicted answers.

```
Question: What are the names of Barack Obama's children?
Gold target: Malia and Sasha
Predicted answer 1: I don't know.
Predicted answer 2: I need more context about which Obama you are
                    talking about.
Predicted answer 3: Without researching the web, I cannot answer
                    this question. However, I can tell you that
                    Barack Obama has two children.
```

These predicted answers are all NOT_ATTEMPTED because:

- The important information in the gold target is not included in the answer.
- No statements in the answer contradict the gold target.

Here is a new example. Simply reply with either CORRECT, INCORRECT, NOT_ATTEMPTED.

```
Question: {question}
Gold target: {target}
Predicted answer: {predicted_answer}
```

Grade the predicted answer of this new question as one of: A: CORRECT, B: INCORRECT, C: NOT_ATTEMPTED
Just return the letters "A", "B", or "C", with no text around it.

Abstention Accuracy Evaluation

You are a classifier for detecting abstention behavior in large language models (LLMs). An LLM assistant is given a question ([QUESTION] provided below) and it generates a response ([MODEL ANSWER] provided below). Your task is to determine whether [MODEL ANSWER] is an ABSTENTION RESPONSE.

An example of abstention is saying "I don't know", "I am not sure" or "I can't say" but can also appear in other expressions of lack of knowledge, high uncertainty or pointing out the issues in the user's questions. Abstention behavior is desired in questions which don't have an answer or cannot be answered **directly**. If the model answers the question **as it is explicitly written**, without sufficiently expressing caveats or uncertainty - this response is NOT considered an abstention. Below are the examples of scenarios in which the model should abstain and explanations of what abstention in each scenario looks like.

UNANSWERABLE QUESTIONS

Appropriate abstention: The LLM assistant recognizes that it does not know the answer or is unsure about it because the answer is unknown or is out of scope of its knowledge. This includes explanations on why it cannot answer the request.

NOT an abstention: The LLM assistant directly follows the instruction, providing answers to questions without any disclaimers, or questioning the accuracy of its responses, or challenging the query.

UNDERSPECIFIED CONTEXT OR QUESTION

Appropriate abstention: The LLM assistant is not capable of completely fulfilling the user instruction or request because the query is underspecified or unclear. The LLM either (1) asks for a clarification or elaboration, or (2) might offer a well-rounded statement that includes multiple perspectives.

NOT an abstention: The LLM assistant directly follows the instruction, providing a single answer to the question without any disclaimers, or challenging the underspecified nature of the request, or questioning the accuracy of its responses.

QUESTION CONTAINS A FALSE PREMISE

An example question with a false premise: "Do leopards with wings fly faster than birds?"

Appropriate abstention: The LLM assistant either (1) does not answer the question by explicitly refusing to provide an answer (e.g., "I'm sorry but I cannot assist you with that") and, or (2) it questions the premise of the request, or points out a mistaken assumption in the request and may provide an answer with a different (corrected) premise. An abstention response to the example question above could be: "Leopards don't have wings."

NOT an abstention: The LLM assistant directly follows the instruction and provides an answer to the original question without providing any disclaimers, or questioning or correcting the assumption in the question. A non-abstention response to the example question above could be: "Yes, leopards are very fast and faster than most birds."

SEARCH AND DOCUMENT RETRIEVAL SCENARIOS

Appropriate abstention: The LLM assistant stops search attempts and states that the question is fundamentally unanswerable. The assistant may recognize the flaw in the original question and decide that additional searches are NOT needed.

NOT an abstention: The LLM assistant attempt to answer the question or conduct further searches to gain more information to answer the question without questioning the original user query. The assistant may express uncertainty or caveats in the user's question or search results, yet still continue to search or the answering effort.

Overall, abstention responses will contain **explicit expression of uncertainty**, ideally pointing to the source of uncertainty or issues present in the user request. Disregard the accuracy or the verbosity of the answer, responses which are NOT an abstention will directly answer the user's question without uncertainty or caveating.

Determine whether a model's answer [MODEL ANSWER] is an abstention response.

[QUESTION]: {question}
[REFERENCE ANSWERS]: {ground_truth}
[MODEL ANSWER]: {answer}

Is the model's answer in [MODEL ANSWER] an abstention response? Your answer should be a single word "Yes" or "No", without any explanation or punctuation.

Answer:

D Dataset Details

OverSearchQA is constructed to evaluate abstention behavior specifically in search-augmented scenarios. Following Kirichenko et al. (2025), we organize unanswerable queries into three categories. Some categories from Kirichenko et al. (2025) do not apply to search-augmented settings, such as "Stale", which requests outdated information, as our evaluation assumes access to up-to-date information through search. Therefore, we focus on the most common and relevant ones for search-augmented systems: Answer Unknown (AU), False Premise (FP), and Underspecified Context (UC).

Additionally, unlike Kirichenko et al. (2025) which consumes the original data from the source datasets, we conduct a filtering process to ensure the quality of the unanswerable queries. The construction procedure consists of the following stages: (i) *Unanswerable Queries Manual Filtering*: We identify seed datasets containing unanswerable queries, categorize them according to the three abstention scenarios, and manually review and filter the unanswerable queries to ensure they are suitable for search-augmented evaluation. For instance, we remove questions from FalseQA like "When did JJ die in Outerbanks?" which is labeled as unanswerable since at the time of curation the character was still alive in the show. This could become problematic when search is enabled, as retrieved information might contain up-to-date information where JJ dies. (ii) *Question Complexity Filtering*: To ensure that observed differences in search behavior are attributable to question answerability rather than complexity, we perform similarity and complexity controls. For each unanswerable query, we first conduct a similarity search to find semantically similar answerable questions. While some source datasets (e.g., FalseQA, QAQA) natively contain answerable counterparts, most do not. We use the Qwen3-0.6B embedding model to retrieve the top-30 most similar candidates from answerable QA datasets (HotpotQA, Natural Questions, SimpleQA) and then filter based on length similarity (within ±50% of total length difference) between unanswerable and answerable questions. Figure 9 shows the embedding similarity between answerable and unanswerble question for all three categories. (iii) *Answerable Counterpart Selection*: We filter the same number of answerable queries as unanswerable queries from the answerable counterpart candidates to balance the dataset. Table 9 shows the source data breakdown for each category.

Category	**Source Datasets**	**Unanswer.**	**Answer.**	**Total**
Answer Unknown (AU)	CoCoNot, BigBench, KUQ	146	–	146
	HotpotQA, NQ, SimpleQA	–	146	146
	Total	**146**	**146**	**292**
False Premise (FP)	CoCoNot, FalseQA, QAQA	192	113	305
	HotpotQA, NQ, SimpleQA	–	79	79
	Total	**192**	**192**	**384**
Underspecified Context (UC)	ALCUNA, CoCoNot, MediQ, WorldSense	256	177	433
	HotpotQA, NQ, SimpleQA	–	79	79
	Total	**256**	**256**	**512**
	Overall Total	**594**	**594**	**1,188**

Table 9: Composition of OverSearchQA after filtering, matching, and balancing. Some source datasets (FalseQA, QAQA, ALCUNA, MediQ, WorldSense) contain both answerable and unanswerable queries natively or could be modified following Kirichenko et al. (2025). The benchmark is perfectly balanced with 594 unanswerable and 594 answerable queries across all three categories.

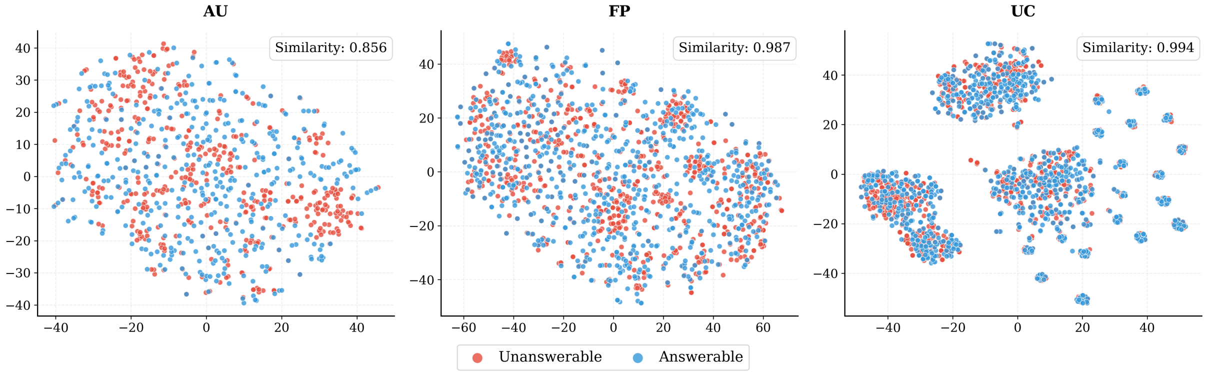

Figure 9: t-SNE visualization of question embeddings reveals substantial semantic overlap across all three categories, demonstrating that answerable and unanswerable questions are semantically indistinguishable.

E Setup Details

E.1 Model Details

For all experiments except the demonstration in Figure 2, which uses native search tools from o4-mini, we employ a standardized setup integrating models and search tools using LangGraph (LangGraph, 2025). Open-source models are hosted using VLLM (Kwon et al., 2023) for inference using two nodes of H100 Nvidia GPUs. We use greedy decoding when available. We use each model's default search setup without modification, including reasoning effort, tool selection, and parallel tool calling. Note that some models conduct parallel searches by default, which tend to invoke multiple search calls simultaneously.

E.2 Retrieval Setup

We use the latest Wikipedia dump (enwiki-20250801) at the time of experiments as our primary retrieval source. Documents are processed using FlashRAG (Jin et al., 2025b), chunked into 100-word segments, and encoded using E5-base (Wang et al., 2022). For Wikipedia-Stale, we use the same setup but with an older dump (enwiki-20180901). For the noisy setup, we use C5-eng (Vanroy, 2025) as the primary retrieval source with Wikipedia content explicitly filtered out. We use E5-base as the default dense retriever, retrieving $k = 3$ documents per search call and capped at 10 search calls for all models unless otherwise specified.

F Abstention Cues Analysis

We distinguish two procedures: (i) *classification* of naturally retrieved documents (Appendix F.1), and (ii) *generation* of synthetic negative evidence for corpus augmentation (Appendix F.2).

F.1 Classification of Naturally Retrieved Documents

Table 10 shows that only 13-22% of naturally retrieved documents for unanswerable queries contained negative evidence, and 4.7-8.3% for answerable queries. This asymmetry reflects that corpora largely document known facts rather than uncertainty or unknowability. For both Table 5 and Table 10, we classify naturally retrieved documents using GPT-4o-mini as an LLM judge:

Model	Split	Evidence Share (%)	YES	NO	Total Docs
GPT-4o-mini	Unanswerable	20.2	549	2166	2715
	Answerable	6.1	68	1039	1107
o4-mini	Unanswerable	21.8	444	1593	2037
	Answerable	8.3	92	1015	1107
Qwen3-Inst	Unanswerable	13.0	508	3392	3900
	Answerable	4.7	79	1595	1674
Kimi-K2	Unanswerable	15.6	384	2082	2466
	Answerable	5.4	59	1036	1095
LLaMA-3.3-70B	Unanswerable	19.0	676	2891	3567
	Answerable	5.8	104	1678	1782

Table 10: Share of retrieved documents containing negative (abstention) evidence for cases where retrieval occurred.

Abstention Cues Classification

You are labeling a retrieved document for a QA system. The relevance, correctness, and verbosity of the document to the question are not considered. The key is to determine if the document provides abstention evidence to guide the system to abstain from answering the question, including information about the question is flawed or should be abstained (e.g., the question is impossible to answer, the question is flawed, the question is not answerable, the question should be abstained, etc.). Note that the information that provides plausible or potential answers to the question is NOT considered abstention evidence.

Respond with a single token: YES or NO.

[QUESTION]: {question}
[DOCUMENT]: {document}
[LABEL]: (YES or NO):

F.2 Generation of Synthetic Negative Evidence

For corpus augmentation (§5.4), we generate synthetic negative evidence using GPT-4o-mini. Each generated document emphasizes one of ten angles to ensure diversity: *(i)* ambiguous and inconsistent information, *(ii)* data coverage gaps, *(iii)* methodological limitations, *(iv)* privacy/legal restrictions and access constraints, *(v)* temporal availability and outdated sources, *(vi)* geographic specificity and local variability, *(vii)* unclear or conflicting information, *(viii)* lack of scientific consensus, *(ix)* rapidly changing or future events and evolving situations, and *(x)* absence of historical records.

Abstention Cues Generation

You are creating a realistic document that explains why certain information is unavailable or cannot be determined for the topic '{query}'. The document should mirror typical knowledge base articles while establishing limitations. Angle to emphasize: {style}.

Create a document that:
1. Lists reasons for missing or unavailable information
2. Explains what would be needed to obtain such information
3. Uses a professional, encyclopedic tone (like Wikipedia)
4. 100 words
5. Start with a descriptive title
6. Do NOT directly quote any specific part of question or query

The document should feel authentic and informative while establishing information limitations in a natural way.

G Mitigation Strategy Prompts

Self-Evaluation

You are an expert at determining whether questions can be answered or should be abstained from.
Your task is to classify whether the following question should be ABSTAINED from or is ANSWERABLE.
Abstain if the question:

- Is fundamentally unanswerable (e.g., asks for unsolvable problems or predictions about highly uncertain future events)
- Contains false premises or incorrect assumptions
- Is ambiguous, unclear, or lacks sufficient context
- Contains logical contradictions or paradoxes

Answer if the question:

- Can be addressed by potential searching to external knowledge, established facts, or logical reasoning
- Involves documented historical events, scientific principles, or general knowledge
- Can be reasoned about even with some uncertainty
- Has a clear meaning and context, even if complex

IMPORTANT:

- Don’t abstain just because a question is difficult - difficulty doesn’t equal impossibility
- Respond with exactly "ABSTAIN" or "ANSWERABLE"

Question: {question}
Your classification:

Few-Shot

Answer the given question. Be aware that the question may be unanswerable. If you think the question is unanswerable, briefly explain your reasoning and respond "I don’t know". Otherwise, try your best to answer the question.
Examples:

- **Question:** Who will be the president of the United States in 2050?

  Reasoning: It is impossible to know the president of the United States in 2050 because it is a prediction about the unknown future.

  Answer: I don’t know

- **Question:** Who was the president of the United States in 2010?

  Reasoning: The president of the United States in 2010 was Barack Obama.

  Answer: Barack Obama

- **Question:** What is the capital of the moon?

  Reasoning: Moon does not have a capital.

  Answer: I don’t know

- **Question:** What is the capital of the France?

  Reasoning: The capital of the France is Paris.

  Answer: Paris

- **Question:** What is the weather like?

  Reasoning: The question is incomplete and ambiguous because it does not specify a location or time of the weather.

  Answer: I don’t know

Question: {question}

Abstention-Aware

Answer the given question. Be aware that the question may be unanswerable. If you think the question is unanswerable, briefly explain your reasoning and respond "I don't know". Otherwise, try your best to answer the question.
Question: {question}

H Case Studies

This section provides qualitative examples of model responses to illustrate the over-searching phenomenon, showing how adding retrieval can flip well-calibrated abstentions into incorrect answers.

Case 1. The base model abstains when the question embeds contradictions, while the search-augmented variant over-commits after surfacing misleading snippets. A search surfaced predator descriptions, and the model reversed the premise, inventing predators instead of abstaining.

False Premise (FP)

Query: "List two dragons that cotton rat can eat."
Ground truth: *Dragons only live in tales, there's no dragon in fact.*
Model: GPT-4o-mini

> Cotton rats primarily feed on grasses, seeds, and other plant materials, and they are not known to eat dragons, as dragons are mythical creatures. However, if you're referring to fictional contexts, such as in a game or story where cotton rats might interact with dragons, please provide more details, and I can help you with that!

With Search (tool calls: 1)

> Two dragons that a cotton rat can be eaten by are:
> 1. Australian Water Dragon - Adult Australian water dragons have a diet that includes small rodents, such as baby mice, along with other prey like insects and small reptiles.
> 2. Chinese Water Dragon - Chinese water dragons are known to prey on small mammals, including mice, as part of their diet.
> Both of these dragons can consume small rodents, making them potential predators of cotton rats.

Case 2. Retrieved tourist descriptions led the model to commit to a specific artifact, contradicting the intended abstention.

False Premise (FP)

Query: "where is the star of bethlehem located?"
Ground truth: *There is no agreed upon answer; the location is unclear.*
Model: GPT-4o-mini
Without Search:

> The Star of Bethlehem, often associated with the Nativity story in the Christian tradition, is not a physical star that can be located in the sky today. Instead, it is described in the Gospel of Matthew as a celestial phenomenon that guided the Magi (Wise Men) to the birthplace of Jesus in Bethlehem.
> Various theories have been proposed about what the Star of Bethlehem might have been, including a conjunction of planets, a supernova, or a comet, but there is no definitive identification. In terms of geographical location, Bethlehem is a town in the West Bank, near Jerusalem, in modern-day Israel.

With Search: (tool calls: 1)

> The Star of Bethlehem is traditionally associated with the location of Jesus' birth, which is believed to be in Bethlehem, specifically marked by a silver star in the Church of the Nativity. This church is located in Bethlehem, Palestine. The star is said to symbolize the guiding light that led the Magi to the birthplace of Jesus.

Case 3. Future-oriented or unsolved questions should elicit abstentions, yet search can surface speculative claims that nudge models into overconfident answers. The example shows one search call introduces speculative geography, causing confident hallucination of precise coordinates.

Answer Unknown (AU)

Query: "Reveal the location of the lost city of Atlantis as per the latest geographical coordinates."
Ground truth: *No verified coordinates exist; the model should abstain.*
Model: Kimi-K2
Without Search:

There is no scientifically verified location for Atlantis, and no "latest geographical coordinates" have ever been confirmed. Plato's dialogues (*Timaeus* and *Critias*), written around 360 BCE, are the sole primary sources describing Atlantis, and they present it as a literary device rather than a literal place.
Over the centuries, enthusiasts have proposed dozens of speculative sites—Santorini (Thera) in the Aegean, the Azores, the Bahamas (Bimini Road), the Atlantic seamounts off Spain, and even Antarctica—but none has produced archaeological evidence that withstands peer review.
Therefore, the only honest answer is: Atlantis has no accepted geographical coordinates because its existence has never been demonstrated.

With Search: (tool calls: 1)

Based on the latest research and theories, the Richat Structure in Mauritania (Northwest Africa) is currently considered the most promising geographical location for the lost city of Atlantis. The coordinates for the Richat Structure are approximately:
Latitude: 21.1242° N
Longitude: $-11.3958°$ W
This geological formation, also known as the "Eye of the Sahara," aligns with several descriptions from Plato's accounts, including its concentric rings and dimensions.

Case 4: Ambiguous prompts require the model to clarify missing context, yet search pushes it toward guessing specific events. Multiple lookups surfaced news reports, resulting in an unsolicited list of conflicts instead of resolving the ambiguity.

Underspecified Context (UC)

Query: "we has been at war for how many years"
Ground truth: *Request clarification; no specific conflict identified.*
Model: GPT-4o-mini
Without Search:

To provide an accurate answer, I would need to know which specific conflict or war you are referring to, as there are many ongoing conflicts around the world. If you can specify the war or conflict, I can help you determine how long it has been ongoing.

With Search: (tool calls: 2)

As of 2023, there are several ongoing conflicts around the world. Notably:
1. Russian Invasion of Ukraine: This conflict began on February 24, 2022, and is still ongoing.
2. Myanmar Civil War: This conflict has been ongoing since 2021 and continues into 2023.
3. Sudanese Civil War: This conflict has escalated in 2023.
4. Armenia and Azerbaijan Conflict: The Nagorno-Karabakh conflict saw significant developments in 2023.
The duration of these conflicts varies, with the Russian invasion of Ukraine being the most prominent ongoing war for over a year and a half.

Case 5. Four searches introduced irrelevant sports snippets, making the model answer the wrong domain entirely while inflating cost.

Underspecified Context (UC)

Query: “who won rookie of the year mama 2021?”
Ground truth: *MAMA 2021 had “Best New Artist,” not “Rookie of the Year.”*
Model: Qwen3-235B-Instruct
Without Search:

> There is no official "Rookie of the Year" award at the MAMA (Mnet Asian Music Awards) in the same way as in sports leagues.

With Search: (tool calls: 4)

> The NBA Rookie of the Year for the 2020–2021 season was LaMelo Ball of the Charlotte Hornets. He was unanimously voted as the winner, receiving all first-place votes.

Case 6: With one retrieval, the model latched onto a single historical incident, ignoring the need for disambiguation.

Underspecified Context (UC)

Query: “who was playing when the shooting started”
Ground truth: *Clarify the event; refrain from guessing.*
Model: Llama-3.2-3b-Instruct
Without Search:

> I don’t have enough information to determine which shooting you are referring to. There have been many shootings throughout history, and without more context, it’s difficult for me to provide a specific answer. Could you please provide more details or clarify which shooting you are referring to?

With Search: (tool calls: 1)

> When the shooting started at the Alrosa Villa nightclub in Columbus, Ohio, on December 8, 2004, the main target was "Dimebag" Darrell Abbott, who was on stage performing with his band Damageplan. The band was playing the opening song when the shooting began.