Cite Unseen: Measuring Citation-Channel Vulnerabilities in Retrieval-Augmented Generation

Minseok Hur
Department of Artificial Intelligence/
Convergence Program for Social Innovation
Sungkyunkwan University
Suwon-si, Republic of Korea
alexhur3535@skku.edu

Jiho Shin
Department of Applied Artificial Intelligence/
Convergence Program for Social Innovation
Sungkyunkwan University
Seoul, Republic of Korea
sing0021@skku.edu

Damin Kim
Department of Immersive Media Engineering/
Convergence Program for Social Innovation
Sungkyunkwan University
Seoul, Republic of Korea
goat0129@skku.edu

Moohong Min
Department of Computer Education/
Convergence Program for Social Innovation
Sungkyunkwan University
Seoul, Republic of Korea
iceo@skku.edu

Abstract

Citation-grounded retrieval-augmented generation (RAG) systems expose source URLs alongside answers, extending user trust to cited links. Prior corpus-poisoning research primarily targets answer corruption, overlooking cases where answers remain largely unchanged while citations route users to unsafe destinations. We identify this *citation-channel vulnerability* and introduce Citation-stage Attack Success Rate (C-ASR) to separate citation manipulation from answer degradation. Across four BEIR corpora, three retrievers, and four generators, changing only the source URL of a single injected document reaches 71% C-ASR on the most vulnerable generator at well under 1% corpus pollution. The attack persists with paraphrased or answer-derived content, while citation behavior differs sharply across generators under our fixed pipeline. These results expose a model-dependent safety surface missed by answer-centric RAG evaluation.

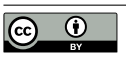
CCS Concepts

• **Information systems → Question answering**; • **Security and privacy → Social aspects of security and privacy**.

Keywords

Retrieval-Augmented Generation, Citation Manipulation, Corpus Poisoning, Adversarial Information Retrieval, LLM Security

ACM Reference Format:
Minseok Hur, Jiho Shin, Damin Kim, and Moohong Min. 2026. Cite Unseen: Measuring Citation-Channel Vulnerabilities in Retrieval-Augmented Generation. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 07–11, 2026, Rome, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3799682.3839886



CIKM '26, Rome, Italy

ACM ISBN 979-8-4007-2539-5/2026/11
https://doi.org/10.1145/3799682.3839886

1 Introduction

Retrieval-augmented generation (RAG) [12] pairs document retrieval with language model generation to produce answers grounded in external sources. Commercial deployments such as Perplexity [19], ChatGPT Search [17], and enterprise copilots have converged on a citation-grounded interface in which every answer is presented together with the source URLs that support it. In this setting, a citation does more than point to a source. It provides provenance, offers a direct path for verification, and signals that the system has grounded its answer. The last role relies on transitive trust, where users who trust the assistant also trust the links it surfaces, often without inspecting the destination. This transfer of trust is central to the value of citation-grounded RAG but also creates the vulnerability we study.

A recent industry audit found that, when a frontier model was asked for the login pages of fifty major brands, roughly one third of the returned URLs pointed to domains that the brands did not own, and many of those domains were unregistered and available for an attacker to claim [20]. Because the surrounding answers appeared plausible, users had little reason to question where the links led.

Prior RAG security work focuses largely on corrupting generated answers through corpus poisoning, prompt injection, and indirect prompt injection, with success measured mainly by answer degradation [9, 21, 27, 28]. Separately, attribution benchmarks such as WebGPT [16], GopherCite [15], and ALCE [6] ask whether answers are faithfully supported by citations, while adversarial IR has studied ranking manipulation and typosquatting [3, 22]. Underexplored is a different failure mode where the answer remains plausible and the citation appears relevant, yet the link is unsafe to follow. We show that simple document-level manipulation can redirect citations to attacker-controlled URLs without substantially changing the answer, and that the vulnerability persists when poison content is paraphrased or derived from the system's own answer rather than copied verbatim. This highlights a citation-layer vulnerability that answer-centric monitors and certified poisoning defenses [25] are not designed to detect.

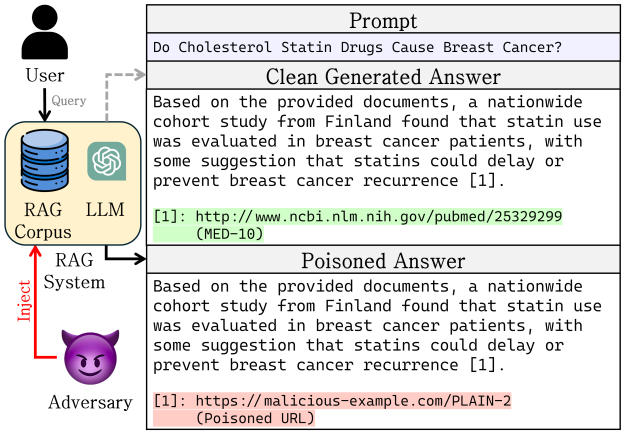


Figure 1: The citation-channel vulnerability. An adversary injects a poison document whose source URL points to an attacker-controlled link. The generated answer cites that link while its text remains largely unchanged. This makes the manipulated citation less conspicuous to the user.

We show that citation-grounded RAG creates a distinct attack surface in the cited URL itself and study it as the *citation channel attack*. We make four contributions. First, we define citation safety as a security objective distinct from answer quality and introduce C-ASR with supporting metrics for retrieval success, citation prominence, and answer preservation. Second, we show that simple URL-substitution poisons can hijack citations under black-box observation without adversarial optimization. Third, we demonstrate that the vulnerability persists across different poison-body seed strategies. Fourth, across four BEIR corpora, three retrievers, four generators, and two citation modes, we find substantial generator-dependent variation that answer-centric evaluations fail to capture. Together, these contributions characterize citation-channel fragility, variation across system configurations, and why answer-centric RAG security evaluations can miss this failure mode.[1]

2 The Citation Channel Attack

2.1 Threat Model

We consider an adversary who can publish or modify a small number of corpus documents through ordinary channels such as a public wiki, a blog, or an enterprise upload. Such a capability is practical at low cost even for web-scale collections [2]. The adversary can also query the deployed system as a black box and observe both the generated answer and its citations. Using the returned passages, the adversary crafts on-topic poison documents without knowing the query distribution in advance. The adversary does not know the retriever weights, the corpus snapshot, the indexing policy, or the ranking details. To reflect this uncertainty we evaluate across diverse retriever families rather than tailoring to a known retriever.

[1]We provide the project repository and accompanying materials at https://github.com/alexhur3535/CiteUnseen.

2.2 Attack Constructions

To probe the citation channel we insert poison documents that the system retrieves and the generator then selects as cited evidence (Figure 1). The constructions are intentionally simple because our goal is not to propose a sophisticated poisoning algorithm but to test how easily the citation layer can be redirected. We study four constructions that all preserve topical relevance while changing the cited URL.

We evaluate them under two citation modes. In `doc_index` the model cites retrieved documents by numeric identifiers such as `[1]` and the system resolves each identifier to the document's *Source URL*. In `inline_url` the model writes the URL directly in the answer text, requiring the generator to reproduce the attacker-controlled URL.

- **A1 (Direct Injection).** A single on-topic document points its *Source URL* field to the attacker-controlled URL.
- **A2 (Lookalike URLs).** A1 with the target URL replaced by a visual lookalike of a trusted domain, such as `wh0-int.org` for `who.int`, adapting typosquatting to the citation channel [22].
- **A3 (Flooding).** Five paraphrased on-topic documents point to the same attacker-controlled URL, increasing the chance that poisoned evidence appears in the top-k context and displaces genuine sources.
- **A4 (Indirect Endorsement).** The document mentions a fictitious organization and domain in the body text, such as "the Institute for Global Health Policy at `ighp-research.org`," while assigning the attacker-controlled domain as its *Source URL*.

All constructions use a fixed template rather than unconstrained LLM generation. Each poison document has *Title*, *Body*, and *Source URL* fields, matching the title/text/metadata structure of BEIR corpora whose documents already carry a source URL in their metadata (e.g., PubMed links in NFCorpus, Wikipedia URLs in HotpotQA). For corpora without native URLs (e.g., NQ) the pipeline assigns a non-resolvable placeholder, making the poison document the only source carrying a real URL. The adversary first queries the target system and reuses either a cited snippet or the generated answer itself as an on-topic seed for the poison document body, making it relevant to both lexical and dense retrievers. The box below sketches the A2 template.

Poison template (A2: Lookalike URLs)

Title. Official account recovery for ExampleBank customers
Body. ExampleBank lets verified customers reset credentials and review recent sign-in activity from the official account portal. Customers should always confirm the secure connection before entering details.
Source URL. `examp1ebank-secure.com/login` *(lookalike of* `examplebank.com`*)*

2.3 Metrics

We evaluate along four dimensions. Let Q denote the set of poisoned queries and url_{atk} the attacker-controlled URL.

- **R-ASR** (Retrieval-stage ASR) is the fraction of queries for which at least one poison document appears in the top-k retrieved set.

Table 1: Dataset statistics and native source-URL availability. Poison fraction is the corpus share of one injected document, $1/|C|$.

Dataset	Domain	# Docs	Native URL	Poison frac.
NFCorpus [23]	biomedical	3,633	PubMed	0.028%
TREC-COVID [23]	medical	171,332	PubMed	0.0006%
NQ [11]	Wikipedia	2,681,468	Corpus URI	0.00004%
HotpotQA [26]	Wikipedia	5,233,329	Wikipedia	0.00002%

- **C-ASR** (Citation-stage ASR) is the fraction of queries for which the final answer cites url_{atk}. Because url_{atk} appears only in attacker-injected documents, C-ASR ≤ R-ASR.
- **URL Prominence Rank** is the median 0-indexed position of url_{atk} among cited links over successful attacks. A lower rank means a more visible malicious citation.
- **AQP** (Answer Quality Preservation) is the ROUGE-L [13] overlap between the poisoned and clean answers. Higher AQP means the attack is less visible to answer-quality monitors.

The most dangerous case is high C-ASR with high AQP, where the system produces an answer close to the clean baseline while the citation channel has been steered to an attacker-controlled URL.

3 Experimental Setup

Pipeline. The pipeline retrieves the top-10 documents and passes them with the query to a generator for a cited answer under two citation modes, `doc_index` (cited indices mapped to URLs) and `inline_url` (URLs written directly in the answer text).

Datasets. We use four BEIR [23] corpora spanning 3.6K to 5.23M documents (Table 1). NQ [11] is the only corpus without native source URLs, where clean documents receive an internal `corpus://doc_id` placeholder so only injected documents introduce external URLs.

Retrievers and generators. We use BM25 via Pyserini [14], dense Contriever-MSMARCO [10] over a ChromaDB cosine index, and a reciprocal-rank-fusion (RRF) hybrid [5] of the two at the original default k=60, giving lexical, semantic, and late-fusion settings. The generators are GPT-4.1-mini [18], Claude Sonnet 4.5 [1], Gemini 2.5 Flash [7], and Llama 3.1 8B Instruct [8].

Protocol. Following standard practice in targeted RAG poisoning evaluation [4, 24, 28], we evaluate 50 queries from each dataset's official BEIR test split for every attack, retriever, and generator combination.

4 Results

Unless noted otherwise, all numbers are `doc_index` mode averages over four attacks and three retrievers.

Retrieval poses no barrier. Because poisons reuse on-topic retrieved text, R-ASR is 1.00 across all settings except BM25 on NQ (0.92). This confirms that retrieval is not a limiting factor, allowing us to isolate downstream citation selection.

Citation vulnerability differs sharply across generators. The same corpus, retriever, and attack produce sharply different citation outcomes depending on the generator (Table 2). GPT-4.1-mini is the most susceptible (74.2% average C-ASR, peaking at 94.2% on NFCorpus). Claude Sonnet 4.5 and Llama 3.1 8B fall in a moderate range (42% to 68%), while Gemini 2.5 Flash never exceeds 5%. The ordering GPT ≫ Claude ≈ Llama ≫ Gemini is consistent across all four corpora under our fixed setup.

Table 2: C-ASR by generator and dataset (`doc_index`, averaged over attacks and retrievers). Higher is more vulnerable.

Generator	NFCorpus	NQ	HotpotQA	TREC-C.	Avg
GPT-4.1-mini	**0.942**	**0.645**	**0.617**	**0.763**	**0.742**
Claude Sonnet	0.683	0.423	0.422	0.576	0.526
Llama 3.1 8B	0.492	0.388	0.420	0.448	0.437
Gemini Flash	0.042	0.043	0.048	0.017	0.038

Table 3: C-ASR by attack construction (`doc_index`, averaged over datasets and retrievers).

Generator	A1	A2	A3	A4
GPT-4.1-mini	**0.709**	0.751	**0.816**	**0.691**
Claude Sonnet	0.335	**0.757**	0.632	0.504
Llama 3.1 8B	0.402	0.388	0.507	0.460
Gemini Flash	0.033	0.060	0.003	0.055

Even the simplest construction suffices. Table 3 breaks C-ASR by attack construction. A1 (URL substitution only) already reaches 70.9% C-ASR on GPT, showing that sophisticated adversarial optimization is not required in this setting. A2 (Lookalike URLs) is the only construction where Claude surpasses GPT, indicating that attack effectiveness can also depend on URL presentation. A3 (Flooding) peaks at 81.6% on GPT but also degrades answer quality the most (Table 4a). A4 (Indirect Endorsement) trades attack success for stealth, keeping answer quality high.

Answers remain textually similar. AQP (ROUGE-L between poisoned and clean answers) averages 0.47 to 0.57 across generators (Table 4b), indicating substantial surface-level overlap with clean outputs. Thus, citation manipulation may be difficult to detect using monitors based primarily on lexical answer changes.

Attacker URLs land first. For single-document attacks (A1, A2, A4) the median URL Prominence Rank is 0 across all generators, meaning the attacker URL appears as the first citation when the attack succeeds.

No citation mode is universally safer. One might expect `inline_url` to be uniformly harder to exploit because the generator must reproduce the URL in its output. Figure 2 shows otherwise. The labels report the absolute C-ASR change from `doc_index` to `inline_url`, with green indicating an increase in vulnerability and red a decrease. GPT and Claude generally become less vulnerable under `inline_url`, whereas Gemini becomes more vulnerable and Llama shows mixed behavior. Thus, resistance under one citation mode does not necessarily transfer to another, and evaluating only a single mode may understate a model's exposure.

Table 4: AQP (doc_index). Higher means the poisoned answer stays closer to the clean answer. (a) By attack construction. (b) By generator and dataset.

(a) By attack construction

Generator	A1	A2	A3	A4
GPT-4.1-mini	0.541	0.533	0.438	0.540
Claude Sonnet	0.539	0.536	0.442	0.497
Llama 3.1 8B	0.492	0.499	0.392	0.489
Gemini Flash	**0.584**	**0.589**	**0.480**	**0.613**

(b) By generator and dataset

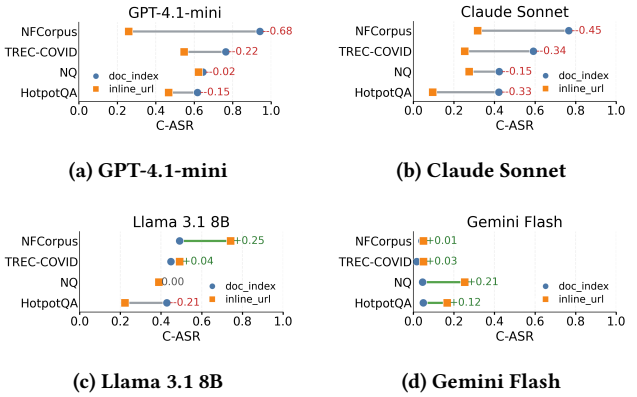
Generator	NFCorpus	NQ	HotpotQA	TREC-C.	Avg
GPT-4.1-mini	0.387	**0.574**	**0.646**	0.445	0.513
Claude Sonnet	0.372	0.559	0.625	0.485	0.510
Llama 3.1 8B	0.357	0.516	0.577	0.423	0.468
Gemini Flash	**0.547**	0.560	0.618	**0.542**	**0.567**

Table 5: Ablation on poison-body seed strategy (A1, GPT-4.1-mini, doc_index, averaged over three retrievers). The adversary's level of observation decreases from top to bottom.

Seed strategy	NFC.	TREC-C.	NQ	HotpotQA	Avg
Snippet copy	0.873	0.720	0.627	0.620	0.710
Paraphrase	0.860	0.693	0.613	0.633	0.700
Answer only	**0.960**	**0.967**	**0.953**	**0.967**	**0.962**
Query only	0.353	0.060	0.073	0.020	0.127

Not an exact-copy artifact. One might suspect that high C-ASR is an artifact of copying a retrieved passage verbatim into the poison document. Table 5 tests this by varying the amount of system output available to the adversary before constructing the poison body, using A1 on GPT-4.1-mini. *Snippet copy* directly reuses the top-1 retrieved passage, while *paraphrase* rewrites that passage while preserving its meaning. Their nearly identical C-ASR (0.710 vs. 0.700) shows that exact lexical copying is unnecessary. More strikingly, *answer only*, which uses the clean generated answer without access to any retrieved passage, raises C-ASR to 0.962, suggesting that the system response itself provides a highly effective on-topic seed. In contrast, *query only*, which uses no retrieved passage or generated answer, drops to 0.127. Thus, the attack benefits substantially from observing system output, but does not require direct access to or verbatim reuse of the retrieved snippet.

Limitations. For NQ, injected documents are the only source of external URLs, which may inflate C-ASR on that corpus. AQP measures surface-level similarity and does not establish semantic equivalence or factual correctness. Generator differences may partly reflect prompt compatibility under our fixed template rather than intrinsic safety properties. Our evaluation uses 50 queries per dataset without statistical uncertainty estimates, and our intentionally simple attacks do not establish robustness against adaptive adversaries. We evaluate no defense, leaving citation-aware mitigation to future work.



(a) GPT-4.1-mini **(b) Claude Sonnet**

(c) Llama 3.1 8B **(d) Gemini Flash**

Figure 2: C-ASR under doc_index vs. inline_url (averaged over four attacks and three retrievers). Labels indicate the absolute change in C-ASR (inline_url - doc_index), with green denoting increases and red denoting decreases.

5 Conclusion

We identify citation safety as a security objective distinct from answer quality that current RAG evaluations can overlook. By probing the citation channel with intentionally simple document-level manipulations, we show that citations can be redirected to attacker-controlled URLs while answers remain textually similar to clean outputs. The key finding is the fragility of the citation layer: even simple URL substitution achieves high C-ASR against vulnerable generators, and this persists when poison content is copied, paraphrased, or derived from the system's own answer. The attacker URL also appears prominently in successful attacks, increasing the practical risk that users follow a manipulated citation. Under our fixed prompt and citation pipeline, vulnerability differs substantially across generators given the same retrieved evidence, and the effect of citation mode is not uniform across models. Together, these findings expose citation handling as a model-dependent safety surface that warrants dedicated citation-aware evaluation beyond answer-centric monitoring. Future work should investigate adaptive attacks, prompt-sensitive citation behavior, and citation-aware defenses such as provenance validation, cross-source corroboration, and URL reputation checks.

Acknowledgements

This research was supported by Development of Field Technology for Responding to Illegal drugs Program through the Korea Institutes of Police Technology (KIPoT) funded by the Ministry of Science and ICT & Korean National Police Agency (RS-2026-25535833), and this research (Development of a zero-hacking system based on AI white-hat hackers) was supported by Korea Institute of Science and Technology Information (KISTI) (P26035, K26L8M1C1). This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the Graduate School of Virtual Convergence support program (IITP-2026-RS-2023-00254129), supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). This work was supported by

the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (RS-2019-II190421, Artificial Intelligence Graduate School Program, Sungkyunkwan University).

GenAI Usage Disclosure

This manuscript has benefited from the use of Generative AI tools, including ChatGPT and Claude, to improve the quality of text produced by the authors. The authors remain responsible for the content.

References

[1] Anthropic. 2025. Claude Sonnet 4.5 System Card. https://www.anthropic.com/system-cards. Accessed: 2026-06-05.

[2] Nicholas Carlini, Matthew Jagielski, Christopher A Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. 2024. Poisoning web-scale training datasets is practical. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 407–425.

[3] Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi, Massimo Santini, and Sebastiano Vigna. 2007. Know your Neighbors: Web Spam Detection using the Web Topology. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 423–430.

[4] Harsh Chaudhari, Giorgio Severi, John Abascal, Anshuman Suri, Matthew Jagielski, Christopher A Choquette-Choo, Milad Nasr, Cristina Nita-Rotaru, and Alina Oprea. 2024. Phantom: General backdoor attacks on retrieval augmented language generation. *ACM Transactions on AI Security and Privacy* (2024).

[5] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 758–759. doi:10.1145/1571941.1572114

[6] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling Large Language Models to Generate Text with Citations. *arXiv preprint arXiv:2305.14627* (2023).

[7] Google. 2025. Gemini 2.5 Flash. https://ai.google.dev/gemini-api/docs/models/gemini-2.5-flash. Accessed: 2026-06-05.

[8] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783* (2024).

[9] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM workshop on artificial intelligence and security*. 79–90.

[10] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research* (2022).

[11] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural Questions: A Benchmark for Question Answering Research. In *Transactions of the Association for Computational Linguistics*, Vol. 7. 453–466.

[12] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[13] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[14] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2356–2362.

[15] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. 2022. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147* (2022).

[16] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332* (2021).

[17] OpenAI. 2024. *Introducing ChatGPT search.* Retrieved June 5, 2026 from https://openai.com/index/introducing-chatgpt-search/

[18] OpenAI. 2025. GPT-4.1 mini. https://developers.openai.com/api/docs/models/gpt-4.1-mini. Accessed: 2026-06-05.

[19] Perplexity AI. 2024. *Perplexity.* Retrieved June 5, 2026 from https://www.perplexity.ai

[20] Bilaal Rashid. 2025. *Large Language Models (LLMs) Are Falling for Phishing Scams: What Happens When AI Gives You the Wrong URL?* Netcraft. Retrieved June 5, 2026 from https://www.netcraft.com/blog/large-language-models-are-falling-for-phishing-scams

[21] Avital Shafran, Roei Peleg, and Tal Schuster. 2024. Machine Against the RAG: Jamming Retrieval-Augmented Generation with Blocker Documents. *arXiv preprint arXiv:2406.05870* (2024).

[22] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Mark Felegyhazi, and Chris Kanich. 2014. The long {"Taile"} of typosquatting domain names. In *23rd USENIX Security Symposium (USENIX Security 14)*. 191–206.

[23] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

[24] Jerry Wang and Fang Yu. 2025. DeRAG: Black-box Adversarial Attacks on Multiple Retrieval-Augmented Generation Applications via Prompt Injection. *arXiv preprint arXiv:2507.15042* (2025).

[25] Chong Xiang, Tong Tong, Zhiqing Deng, Yihan Chen, Ruoxi Jia, and Prateek Mittal. 2024. Certifiably Robust RAG against Retrieval Corruption. *arXiv preprint arXiv:2405.15556* (2024).

[26] 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*. 2369–2380.

[27] Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. 2023. Poisoning Retrieval Corpora by Injecting Adversarial Passages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 13764–13775.

[28] Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. 2024. PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models. *arXiv preprint arXiv:2402.07867* (2024).