

Why Retrieval Doesn't Cure Everything: A Review of Hallucination in Retrieval-Augmented Generation

Sanchita H.¹; Skandamahima V. M.²; Akshitha Katkeri³

^{1,2,3}B N M Institute of Technology

Publication Date: 2026/09/24

Abstract: Retrieval-augmented generation (RAG) was introduced as a direct remedy for hallucination in large language models: instead of answering purely from parameters, the model conditions on passages retrieved from an external corpus. In practice, RAG lowers but does not close the hallucination gap, detailed corpora such as RAGTruth and RAGBench show hallucination rates that still range from about 1% to over 20% of responses depending on domain, retriever quality, and model family. This paper reviews the literature on why RAG systems continue to hallucinate even when correct evidence is available in context, organizes the reported causes into a five-part taxonomy (retrieval failure, conflicting evidence, unfaithful generation, over-reliance on parametric memory, and stale source content), and surveys the detection methods (NLI-based faithfulness scoring, LLM-as-judge, internal-state probing, evidence-graph consistency) and mitigation strategies (corrective retrieval, self-reflective generation, verification loops, faithfulness fine-tuning) proposed against each cause. We synthesize benchmark statistics across RAGTruth, RAGBench, and HalluRAG, and close with open challenges: the absence of a standardized evaluation protocol, the compounding of hallucination in multi-hop and agentic RAG, and the persistent gap between embedding-based detectors and reliable detection under real, model-generated hallucinations.

Keywords: Retrieval-Augmented Generation, Hallucination, Faithfulness, Factuality Detection, Ragtruth, Evaluation Benchmarks.

How to Cite: Sanchita H.; Skandamahima V. M.; Akshitha Katkeri (2026) Why Retrieval Doesn't Cure Everything: A Review of Hallucination in Retrieval-Augmented Generation. *International Journal of Innovative Science and Research Technology*, 11(9), 1063-1068. <https://doi.org/10.38124/ijisrt/26sep724>

I. INTRODUCTION

Large language models (LLMs) are fluent enough that an incorrect answer and a correct one can be indistinguishable on the surface. This failure mode, confident output that is factually wrong, unsupported by any source, or outright fabricated, is generally termed hallucination, and it is the single largest obstacle to deploying LLMs in settings such as medicine, law, and enterprise search, where an unverified claim can cause real harm.

Retrieval-augmented generation was proposed by Lewis et al. (2020) precisely as a structural fix: ground the model in retrieved, external passages rather than relying solely on parametric memory learned during pretraining. The intuition is simple, if the model can see the actual source text, it should not need to invent an answer. Adoption of this idea has been rapid; production chat systems, enterprise search tools, and coding assistants now default to some form of retrieve-then-generate pipeline.

The premise of this review is that RAG does reduce hallucination on average, but does not eliminate it, and the residual hallucinations are not random noise, they cluster around identifiable failure points in the retrieval-and-

generation pipeline. This is not a solved problem; it is an active sub-field with its own benchmarks, taxonomies, and detection methods, several of which are still being revised as recently as 2026. Because this is a review and not a new-methods paper, its contribution is to organize this scattered literature into a single taxonomy of causes, map that taxonomy against the corresponding detection and mitigation methods, and consolidate the benchmark statistics that are otherwise spread across a dozen separate corpora.

II. BACKGROUND

➤ RAG Architecture

A standard RAG pipeline has three stages. A retriever encodes the query and searches a corpus (dense vector index, sparse BM25 index, or a hybrid of both) for the top-k most relevant passages. An optional ranker re-scores or filters these candidates for precision. A generator, the LLM then conditions its output on the concatenation of the query and the retrieved passages. Each stage introduces its own error surface: the retriever can miss the right document, the ranker can discard it, and the generator can ignore or misread it even when it is present.

➤ A Taxonomy of Hallucination Types

The hallucination literature broadly separates intrinsic hallucination, where the output directly contradicts the source content, from extrinsic hallucination, where the output cannot be verified against the source at all (it is neither confirmed nor denied by it). A second, orthogonal split distinguishes factual hallucination, the claim is false relative to the world, from faithfulness hallucination, the claim is unsupported by the specific context the model was given, whether or not it happens to be true. RAGTruth operationalizes this with a four-way span-level taxonomy: evident conflict, subtle

conflict, evident baseless information, and subtle baseless information, which we adopt as a reference vocabulary throughout this paper.

III. TAXONOMY OF CAUSES: WHY RAG STILL HALLUCINATES

Grounding a model in retrieved evidence removes one source of error, pure invention from an ungrounded prior, but introduces or leaves untouched five other failure points, summarized in Table 1 and discussed below.

Table 1 A Five-Part Taxonomy of why RAG Pipelines Still Hallucinate, with Supporting Evidence from the Literature.

Cause	Mechanism	Representative evidence
Retrieval failure	Retriever returns irrelevant, incomplete, or low-precision passages; the generator fills gaps with unsupported content	Retrieval quality is repeatedly identified as the dominant lever on downstream faithfulness across RAG surveys
Conflicting evidence	Retrieved passages disagree with each other or with the model's parametric knowledge, and the model lacks a principled way to arbitrate	Motivates verification-loop methods such as Self-RAG and CRAG, which explicitly grade retrieved evidence before use
Unfaithful generation	Correct evidence is retrieved, but the generator paraphrases loosely, drops qualifiers, or drifts from the source	RAGTruth's span-level taxonomy separately labels “evident conflict” and “subtle conflict” hallucinations of exactly this kind
Over-reliance on parametric memory	The model answers from memorized training data and effectively ignores the retrieved context	Sycophancy- and premise-confirming-retrieval studies show internal-consistency checks can validate an answer that is consistent with poisoned or ignored context
Stale or incorrect source content	The retrieved document itself is outdated or wrong, so a faithful summary is still a wrong answer	Domain corpora such as product manuals and financial filings are especially exposed, since they date quickly relative to model deployment windows

Retrieval failure is consistently reported as the largest single driver: if the retriever's top-k results simply do not contain the answer, the generator is being asked to answer an unanswerable question, and it will often guess rather than abstain. Conflicting evidence arises when multiple retrieved passages disagree, or when a passage disagrees with what the model already “believes” from pretraining; the model has no principled arbitration mechanism unless one is explicitly built in, which is the motivation behind verification-loop methods such as Self-RAG and Corrective RAG (CRAG). Unfaithful generation is the case RAGTruth's annotation scheme was built to isolate, the right document was retrieved, but the generated text still drifts from it, whether through loose paraphrase or an added unsupported detail. Over-reliance on parametric memory is closely related to sycophancy: recent work on premise-confirming retrieval shows that internal-consistency checks can be fooled into approving an answer that is merely consistent with misleading context, rather than actually correct. Finally, stale or incorrect source content is a reminder that RAG's faithfulness guarantee is only as good as the corpus, a perfectly faithful summary of an outdated manual is still a wrong answer to the user's real-world question.

IV. DETECTION METHODS

A parallel literature has grown around detecting hallucination in RAG outputs, both for evaluation and as a component inside mitigation pipelines. Five broad families recur:

- Retrieval-grounded consistency / entailment checking, treat the retrieved context as a premise and the generated claim as a hypothesis, and score whether the context entails, contradicts, or is neutral toward the claim, typically using an NLI model.
- LLM-as-judge, prompt a separate (often stronger) LLM to label adherence, relevance, and completeness; RAGBench used GPT-4 with chain-of-thought prompting to reach 93–95% example- and span-level agreement with human annotators on the DelucionQA subset.
- Fine-tuned span detectors, train a smaller model directly on annotated hallucination spans; RAGTruth's Llama-based detector achieves competitive response- and span-level detection and, when used to guide response sampling, cuts hallucination rates by roughly 43–51%.
- Internal-state and embedding-based probing, use the generator's own hidden states, attention patterns, or output embeddings as a signal, as in HalluRAG. However, a 2026 conformal-prediction study found embedding-space detectors collapse to near-random accuracy (about 50%) on real, model-generated hallucinations from RAGTruth, even though they look strong on synthetic hallucination benchmarks, a caution against trusting detector numbers reported only on easy, synthetic test sets.
- Evidence-graph and retromorphic consistency methods, more recent structural approaches that build a graph over retrieved evidence or run the generation process backward to check consistency; one such hierarchical verification method raised answer-level detection F1 from 0.424 to 0.776 on a re-annotated RAGTruth++ split.

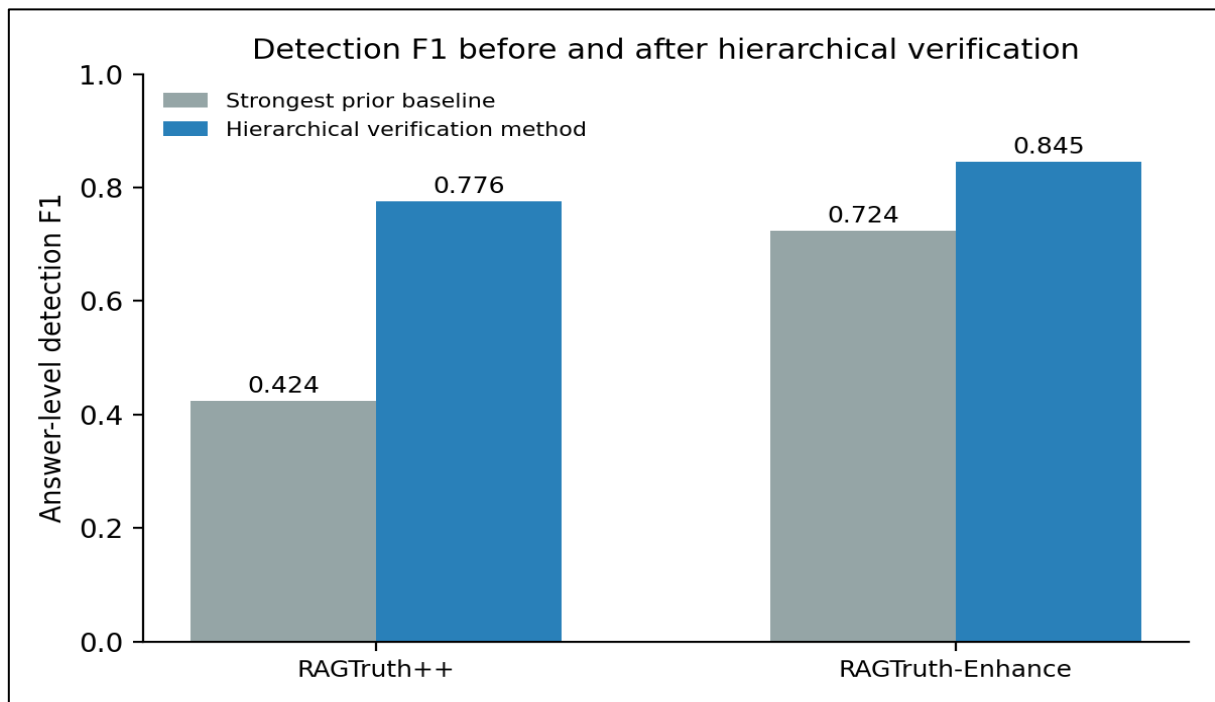


Fig 1 Answer-Level Detection F1 before and after a Hierarchical Verification Pass, Evaluated on Two RAGTruth-Derived Re-Annotations (Data from Retromorphic Testing with Hierarchical Verification, 2026).

A cautionary finding for this whole detection literature: detector accuracy reported on synthetic hallucination benchmarks does not transfer to real, model-generated hallucinations. A 2026 conformal-prediction study (the “Semantic Illusion” analysis) evaluated the same embedding-based detector across four datasets and found a sharp

collapse, shown in Figure 2, near-ceiling accuracy on synthetic corpora such as Natural Questions and WikiBio, but chance-level accuracy on HaluEval and RAGTruth, where the hallucinations are naturally produced by real LLMs rather than synthetically inserted.

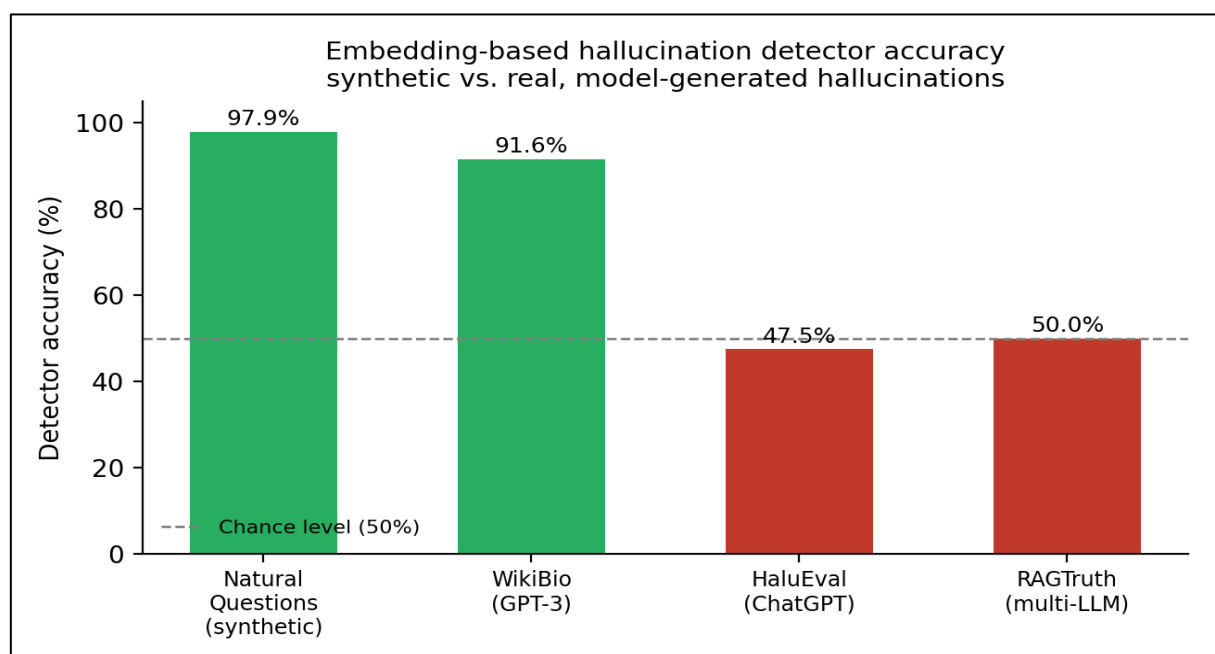


Fig 2 Embedding-Based Detector Accuracy on Synthetic Hallucination Benchmarks Versus Real, Model-Generated Hallucinations (Data from the Semantic Illusion: Certified Limits of Embedding-Based Hallucination Detection in RAG Systems, 2026).

V. MITIGATION METHODS

Mitigation work targets the causes in Table 1 fairly directly. Table 2 groups the main method families with a representative system for each.

Table 2 Mitigation Method Families Mapped Against Representative Systems.

Method family	Representative system	Core idea
Retrieval improvement	Query rewriting, multi-hop / active retrieval (FLARE, RQ-RAG)	Reformulate or re-time queries so retrieval itself surfaces more relevant evidence before generation begins
Self-reflective generation	Self-RAG (Asai et al., 2024)	Special reflection tokens let the model decide when to retrieve and whether its own draft is supported by context
Corrective retrieval	CRAG (Yan et al., 2024)	A lightweight evaluator grades retrieved documents and triggers web search or re-retrieval when confidence is low
Verification / counter-evidence search	Chain-of-Verification; FVA-RAG	Generate a draft, then explicitly search for evidence that would falsify it before finalizing the answer
Fine-tuning for faithfulness	RAFT; RAGTruth-tuned Llama detectors	Train directly on context–response pairs with gold hallucination spans so the model learns to stay grounded
Self-refinement without retrieval change	Iterative self-critique passes	Ask the model to re-check its own draft against context; empirically more consistent than adding RAG alone in some domains

Two points from the empirical literature are worth flagging because they complicate a naive “more RAG is always better” narrative. First, effect sizes are often modest and method-dependent: a 2026 study of package-name hallucination in code generation found that adding RAG lowered hallucination rates by only 1–2 percentage points on average and was actually less effective than simple self-refinement prompting, which delivered a 10–15% relative

reduction, as shown in Figure 3 across three code models. Second, verification-based methods that check only internal consistency between the answer and the retrieved context can be defeated by retrieval that is itself misleading or premise-confirming — a distinction the FVA-RAG line of work makes by explicitly searching for counter-evidence (anti-context) rather than only checking agreement with whatever was retrieved.

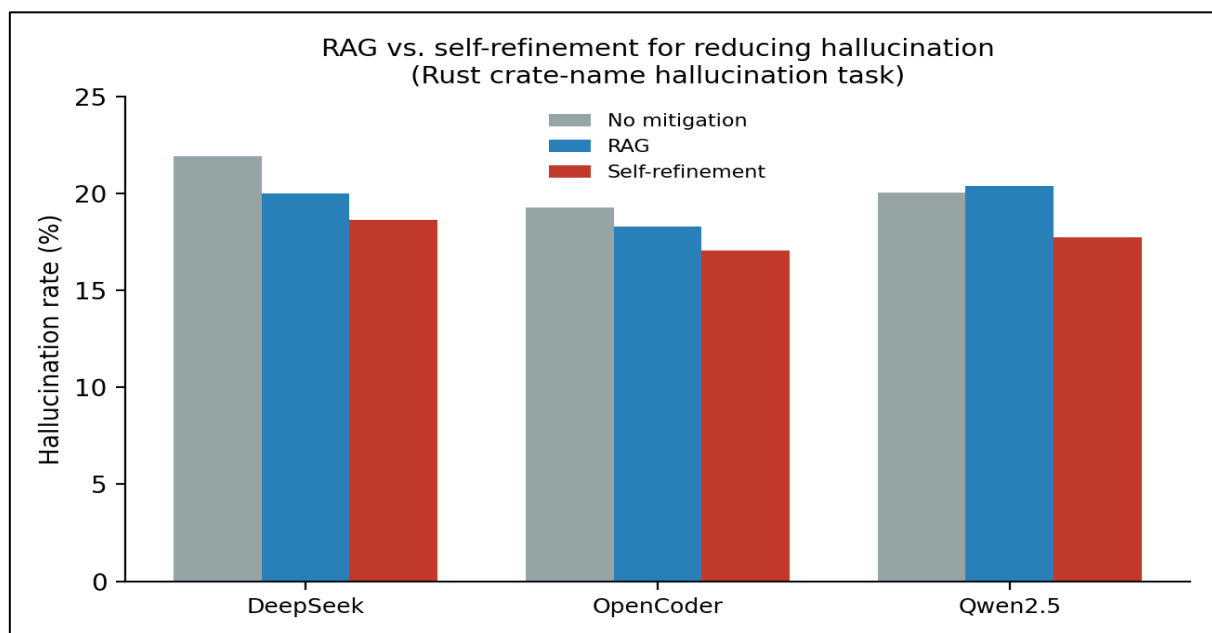


Fig 3 RAG Versus Self-Refinement Prompting for Reducing Hallucination Rate, Evaluated on a Rust Crate-Name Hallucination Task Across Three Code-Generation Models (Data from when LLMs Invent Rust Crates, 2026).

VI. EVALUATION AND BENCHMARKS

Table 3 Summarizes the Benchmarks Most Commonly used to Quantify RAG Hallucination, and Figure 2 Visualizes how Sharply the Reported Hallucination Rate Varies by Domain Even within a Single Benchmark Suite.

Table 3 Benchmarks Used to Quantify and Detect Hallucination in RAG Pipelines.

Benchmark	Scale	Annotation granularity	Key reported statistic
RAGTruth (Niu et al., 2024)	18,000 responses, 6 LLMs	Word/span level, 4-class taxonomy	Fine-tuned Llama detector cut downstream hallucination by 42.9%–51.0% when used for response sampling
RAGBench (2024)	100k+ QA pairs, 5 domains	Example + span level (Adherence, Relevance, Utilization)	Hallucination share ranged 1% (CUAD, FinQA, TAT-QA) to 16% (CovidQA)
HalluRAG (2024)	19,731 annotated sentences	Sentence level, internal-state probing	LLaMA-2-7B $\approx$ 21% hallucination rate vs. Mistral-7B $\approx$ 10%; varies sharply with prompt template
RAGTruth++ / RAGTruth-Enhance (2026 re-annotation)	408 / 2,675 samples	Dual-annotator re-labelling of RAGTruth	Re-annotation found 1.68 $\times$ more hallucinated spans than the original labels

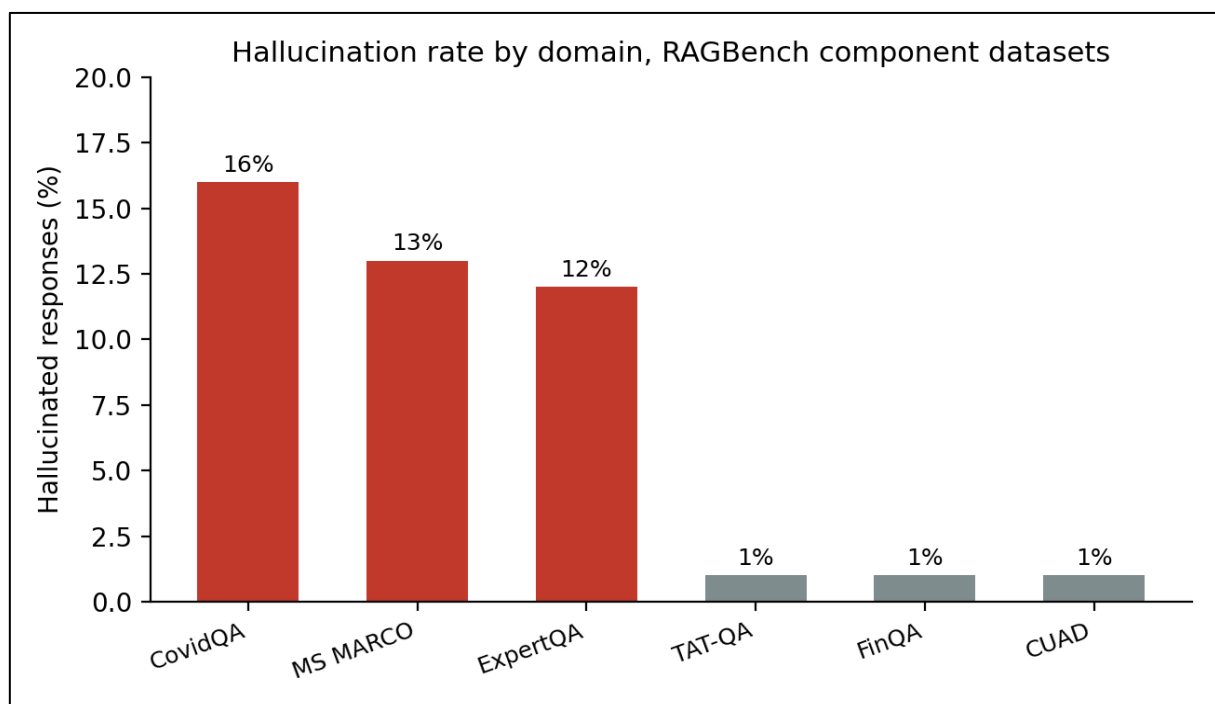


Fig 4 Hallucination Rate by Domain Across RAGBench's Component Datasets. Open-Domain, Fast-Moving Corpora (CovidQA, MS MARCO) Show Substantially Higher Rates than Closed, Structured-Document Domains (CUAD, FinQA, TAT-QA).

Two evaluation-methodology findings are worth highlighting for anyone designing new experiments in this space. First, benchmark labels themselves are not fully reliable: a 2026 re-annotation of RAGTruth's evaluation split found 1.68 times more hallucinated spans than the original human labels contained, which means detector performance numbers computed against the original labels likely understate the true difficulty of the task. Second, LLM-based annotation is converging toward, but has not fully matched, human judgment — a recent adjudicated comparison on QAGS-C and SummEval found human–LLM triple agreement improved by 6–8 percentage points only after a second round of cross-checked human adjudication, implying that single-pass LLM-as-judge labels used in many papers likely carry non-trivial noise.

VII. OPEN CHALLENGES AND FUTURE DIRECTIONS

- Lack of a standardized evaluation protocol — hallucination rate, span-level F1, and entailment-based scores are not directly comparable across papers, and re-annotation studies show even “gold” labels shift substantially under stricter review.
- Multi-hop and agentic RAG — when a system chains several retrieval-and-reasoning steps, an early hallucination can propagate and compound through later steps in ways single-turn benchmarks do not capture.
- Adversarial and premise-confirming retrieval — retrieval that is misleading by construction (sycophantic or poisoned context) breaks internal-consistency-only

verification, motivating counter-evidence-seeking methods as a newer, less mature line of work.

- Real-time and streaming RAG — corpora that update continuously make “stale source content” a moving target rather than a static benchmark property.
- Domain-specific failure modes — legal, medical, and financial corpora each have distinct error costs and document structures (e.g., FinQA and CUAD show far lower raw hallucination rates than open-domain QA, but the cost of a single error is much higher), suggesting domain-specific evaluation thresholds rather than one global standard.

VIII. CONCLUSION

Retrieval-augmented generation measurably reduces hallucination relative to closed-book generation, but the reviewed evidence is consistent on one point: retrieval is necessary, not sufficient. Hallucination in RAG systems is not a single bug but a family of failures distributed across the retriever, the evidence itself, and the generator's faithfulness to what it was given. The detection and mitigation literature has matured quickly — from simple NLI checks to self-reflective generation, corrective retrieval, and graph-based consistency verification — but benchmark re-annotation work published as recently as 2026 shows that even our measurement of the problem is still being corrected. Future systems will likely need to combine better retrieval, explicit uncertainty and abstention, and verification loops that check for counter-evidence rather than mere internal consistency, evaluated against benchmarks whose labels are held to a higher standard than a single annotation pass.

REFERENCES

- [1]. P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.
- [2]. Y. Niu et al., “RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented generation models,” *arXiv:2401.00396*, 2024.
- [3]. L. Huang et al., “A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions,” *ACM Transactions on Information Systems*, vol. 43, no. 2, 2025.
- [4]. Z. Ji et al., “Survey of hallucination in natural language generation,” *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.
- [5]. RAGBench component study, “RAGBench: Explainable benchmark for retrieval-augmented generation systems,” *arXiv:2407.11005*, 2024.
- [6]. HalluRAG, “Detecting closed-domain hallucinations in RAG applications using an LLM's internal states,” *arXiv:2412.17056*, 2024.
- [7]. A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, “Self-RAG: Learning to retrieve, generate, and critique through self-reflection,” in *Proc. ICLR*, 2024.
- [8]. S. Yan et al., “Corrective retrieval augmented generation,” *arXiv:2401.15884*, 2024.
- [9]. S. Dhuliawala et al., “Chain-of-verification reduces hallucination in large language models,” in *Findings of ACL*, pp. 3563–3578, 2024.
- [10]. “Retromorphic testing with hierarchical verification for hallucination detection in RAG,” *arXiv:2603.27752*, 2026.
- [11]. “The semantic illusion: Certified limits of embedding-based hallucination detection in RAG systems,” *arXiv:2512.15068*, 2026.
- [12]. “FVA-RAG: Falsification-verification alignment for mitigating sycophantic hallucinations,” *arXiv:2512.07015*, 2026.
- [13]. “When LLMs invent Rust crates: An empirical study of hallucination patterns and mitigation,” *arXiv:2606.08444*, 2026.
- [14]. “Do benchmarks underestimate LLM performance? Evaluating hallucination detection with LLM-first human-adjudicated assessment,” *arXiv:2605.08462*, 2026.
- [15]. S. Es, J. James, L. Espinosa Anke, and S. Schockaert, “RAGAS: Automated evaluation of retrieval augmented generation,” in *Proc. EACL System Demonstrations*, pp. 150–158, 2024.
- [16]. J. Chen, H. Lin, X. Han, and L. Sun, “Benchmarking large language models in retrieval-augmented generation,” in *Proc. AAAI Conf. on Artificial Intelligence*, vol. 38, pp. 17754–17762, 2024.
- [17]. S. Min et al., “FActScore: Fine-grained atomic evaluation of factual precision in long form text generation,” in *Proc. EMNLP*, pp. 12076–12100, 2023.
- [18]. T. Vu et al., “FreshLLMs: Refreshing large language models with search engine augmentation,” in *Findings of ACL*, pp. 13697–13720, 2024.
- [19]. N. F. Liu et al., “Lost in the middle: How language models use long contexts,” *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 2024.
- [20]. V. Adlakha, P. BehnamGhader, X. H. Lu, N. Meade, and S. Reddy, “Evaluating correctness and faithfulness of instruction-following models for question answering,” *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 681–699, 2024.