Attribution Techniques for Mitigating Hallucinated Information in RAG Systems: A Survey

Yuqing Zhao
Nanyang Technological University, Singapore
ZHAO0522@e.ntu.edu.sg

Ziyao Liu
Nanyang Technological University, Singapore
liuziyao@ntu.edu.sg

Yongsen Zheng†
Nanyang Technological University, Singapore
yongsen.zheng@ntu.edu.sg

Kwok-Yan Lam
Nanyang Technological University, Singapore
kwokyan.lam@ntu.edu.sg

***Abstract*—Large Language Models (LLMs) systems play a critical role in modern AI, demonstrating strong performance across various tasks. However, LLM-generated responses often suffer from hallucinations, unfaithful statements lacking reliable references. Retrieval-Augmented Generation (RAG) frameworks enhance LLM responses by incorporating external references but also introduce new forms of hallucination due to complex interactions between the retriever and generator. To address these challenges, researchers have explored attribution-based techniques that ensure responses are verifiably supported by retrieved content. Despite progress, a unified pipeline for these techniques, along with a clear taxonomy and systematic comparison of their strengths and weaknesses, remains lacking. A well-defined taxonomy is essential for identifying specific failure modes within RAG systems, while comparative analysis helps practitioners choose appropriate solutions based on hallucination types and application context. This survey investigates how attribution-based techniques are used within RAG systems to mitigate hallucinations and addresses the gap by: (i) outlining a taxonomy of hallucination types in RAG systems, (ii) presenting a unified pipeline for attribution techniques, (iii) reviewing techniques based on the hallucinations they target, and (iv) discussing strengths and weaknesses with practical guidelines. This work offers insights for future research and practical use of attribution techniques in RAG systems.**

***Index Terms*—Large Language Models, Attribution Techniques, Retrieval-Augmented Generation(RAG), Hallucinated Information, Hallucination Mitigation.**

I. Introduction

Large Language Models (LLMs) have become foundational in natural language processing tasks such as question answering, summarization, translation, and dialogue. Trained on extensive text corpora, they generalize well and generate fluent, coherent, and contextually appropriate responses. However, they face a major limitation: hallucination—the production of factually incorrect, unsupported, or misaligned outputs—which undermines the reliability of LLM-based systems, especially in high-stakes domains such as healthcare, law, and education.

Retrieval-Augmented Generation (RAG) addresses this by grounding LLM outputs in external knowledge. By retrieving relevant references, RAG improves factual accuracy and interpretability, enabling users to trace responses to supporting evidence. Yet RAG introduces new challenges: retrieval may return outdated or irrelevant information, misleading the generator and producing new hallucinations. Mismatches between retriever and generator can propagate errors and compromise system reliability.

Recent work explores attribution-based methods to reduce hallucination in RAG systems, including improving query formulation, reranking retrieved documents, designing prompts that encourage grounded generation, and applying post-hoc corrections. However, a clear understanding of which techniques address specific hallucination types—especially those arising from retriever–generator interactions—remains limited.

Comparison with related surveys. Existing surveys on hallucination either focus on LLMs broadly or do not connect mitigation techniques to the specific causes of hallucination in RAG systems. Surveys such as [1]–[4] analyze hallucination in general LLMs but overlook RAG-specific error sources. Others [1], [5]–[7] discuss attribution or RAG development but do not provide a taxonomy linking hallucination types to corresponding mitigation strategies. This survey fills the gap by focusing on hallucinations within RAG and organizing attribution methods around retriever–generator interactions.

Summary of contributions.

1) We propose a taxonomy of hallucination types that arise from retriever–generator interactions in RAG systems.

2) We introduce a unified attribution-based mitigation pipeline with four modular components: Query Refining (T1), Reference Identification (T2), Prompt Engineering (T3), and Response Correction (T4), as shown in Fig.1.

3) We map each module to the hallucination types they best mitigate, providing guidance for selecting techniques based on the source of error.

4) We present a comparative analysis of attribution methods in terms of effectiveness, computational efficiency, and application domains, and discuss usage strategies, trade-offs, and open challenges in practical deployment.

† Corresponding author.

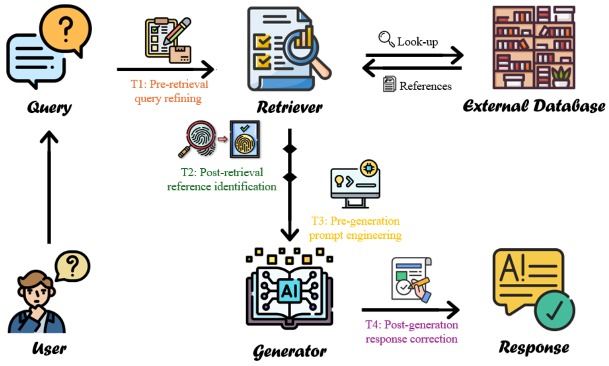
Query
T1: Pre-retrieval query refining
Retriever
Look-up
References
External Database
T2: Post-retrieval reference identification
T3: Pre-generation prompt engineering
User
Generator
T4: Post-generation response correction
Response

Fig. 1. A Unified Pipeline of LLM Attribution

II. A Unified Pipeline of LLM Attribution

The LLM attribution pipeline integrates LLMs with external knowledge to generate responses grounded in credible references. Given a user query q, the retriever returns relevant references R', and the generator produces a response $A = \mathcal{M}(q, \{r\}), \forall r \in R'$. The pipeline includes four modules (Figure 1): Pre-retrieval Query Refining (T1), Post-retrieval Reference Identification (T2), Pre-generation Prompt Engineering (T3), and Post-generation Response Correction (T4). These components improve efficiency, accuracy, and usability without modifying model architecture or requiring fine-tuning.

T1: Refines the input query q to $q' = \mathcal{T}(q)$ for retrieval.

T2: Selects the most relevant references, yielding $R_{\text{final}} = \mathcal{V}(R')$.

T3: Constructs a structured prompt $P = \mathcal{S}(q', R_{\text{final}})$.

T4: Reviews and adjusts the generated response A_{gen} using $A_{\text{corrected}} = \mathcal{C}(q', A_{\text{gen}}, R_{\text{final}})$.

To illustrate the end-to-end interaction, consider the query "Who is the current CEO of OpenAI?". T1 may introduce temporal constraints, T2 prioritize recent authoritative sources, T3 encourage evidence-grounded prompting, and T4 verify and correct unsupported claims, together illustrating how these techniques can operate jointly within an end-to-end pipeline.

III. Taxonomy of Attribution for Mitigating Hallucinations

Hallucination occurs when an LLM generates responses that are factually incorrect, unsupported, or misaligned with the query or references. In RAG systems, hallucinations may arise from retrieval or generation. This section presents a taxonomy of hallucination types specific to RAG, based on retriever–generator interactions (Figure 2), providing a framework for linking attribution methods to each hallucination type.

A. Regulating Overconfidence Hallucination

Overconfidence hallucination occurs when an LLM expresses uncertain or incorrect information with excessive certainty. Two attribution-based techniques help mitigate this: T3 and T4.

T3 Overconfidence emerges when responses display unwarranted certainty. Adding hedging terms (e.g., likely, possibly) reduces this. T3 incorporates such cues into prompts. Verbal Uncertainty Calibration (VUC) [12] inserts hedging instructions to align response confidence with that of retrieved references, using a verbal uncertainty feature in the representation space.

T4 Overconfidence may also result from over-reliance on particular references or from mismatched certainty between generated content and retrieved evidence. Post-generation correction addresses this by adjusting confidence [8], [9], [11] and limiting dominance of specific references [10]. RLKF [11] uses a reward model to penalize false certainty. SAPLMA [8] employs a classifier to detect confidently stated falsehoods and trigger regeneration. Iterative Feedback Learning (IFL) [9] evaluates correctness and citation quality to refine responses and diversify reference use. SelfCheckGPT [10] measures consistency across multiple generated outputs, selecting the version most supported by retrieved references to identify unsupported overconfident claims.

B. Managing Outdatedness Hallucination

Outdatedness hallucination occurs when an LLM generates responses that were once correct but are now obsolete. Attribution-based solutions primarily rely on T1 and T2.

T1 Outdatedness often arises from ambiguous queries or missing temporal cues that lead to retrieving old information. Query refining improves precision and timeliness [13], [14]. SmartBook [14] adds temporal keywords (e.g., "past two weeks") to restrict retrieval to recent sources. WebCPM [13] enhances WebGPT through iterative refinements such as synonym substitution and paraphrasing to obtain fresher references.

T2 Outdatedness also results from selecting references that are no longer current. Post-retrieval identification methods address this extensively [15]–[22]. DPR [16] evaluates timeliness using relevance, frequency, and source reliability, forming a foundation for RAG systems [19], [35]. Later approaches enhance retrieval freshness: LLM-Augmenter [15] applies RLHF for date-specific retrieval, CoDA [20] prioritizes timely over popular references, REALM [21] updates retrieval dynamically, and FDP [22] estimates temporal validity. WebBrain [18] maintains an updated Wikipedia-based database, while WebGPT [17] simulates real-time web searches, a strategy now common in systems such as ChatGPT, DeepSeek, and Anthropic.

C. Alleviating Unverifiability Hallucination

Unverifiability hallucination arises when an LLM produces responses without sufficient supporting evidence. Attribution-based methods mitigate this by using T2 and T4.

T2 Unverifiability often results from loosely related or weakly aligned references. Post-retrieval identification ensures that only verifiable, well-supported evidence informs the final answer, using either reranking [23]–[26] or finer-grained reference selection [27]–[30]. LLM-assisted reranking methods such as REPLUG [23], AAR [24], and SELF-RAG [26] classify retrieved documents as fully, partially, or unsupported

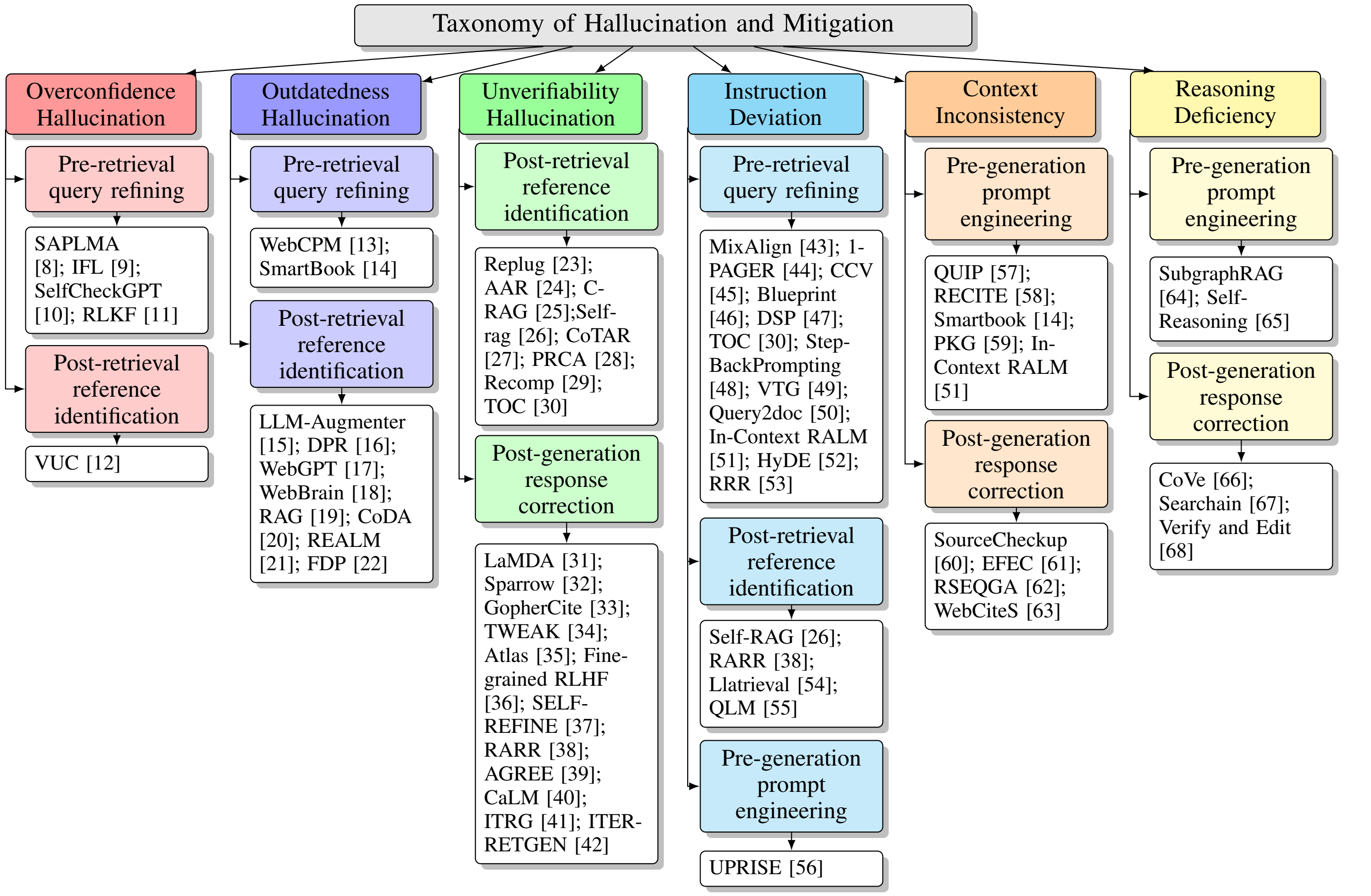
Taxonomy of Hallucination and Mitigation
Overconfidence Hallucination
Pre-retrieval query refining
SAPLMA [8]; IFL [9]; SelfCheckGPT [10]; RLKF [11]
Post-retrieval reference identification
VUC [12]
Outdatedness Hallucination
Pre-retrieval query refining
WebCPM [13]; SmartBook [14]
Post-retrieval reference identification
LLM-Augmenter [15]; DPR [16]; WebGPT [17]; WebBrain [18]; RAG [19]; CoDA [20]; REALM [21]; FDP [22]
Unverifiability Hallucination
Post-retrieval reference identification
Replug [23]; AAR [24]; C-RAG [25];Self-rag [26]; CoTAR [27]; PRCA [28]; Recomp [29]; TOC [30]
Post-generation response correction
LaMDA [31]; Sparrow [32]; GopherCite [33]; TWEAK [34]; Atlas [35]; Fine-grained RLHF [36]; SELF-REFINE [37]; RARR [38]; AGREE [39]; CaLM [40]; ITRG [41]; ITER-RETGEN [42]
Instruction Deviation
Pre-retrieval query refining
MixAlign [43]; 1-PAGER [44]; CCV [45]; Blueprint [46]; DSP [47]; TOC [30]; Step-BackPrompting [48]; VTG [49]; Query2doc [50]; In-Context RALM [51]; HyDE [52]; RRR [53]
Post-retrieval reference identification
Self-RAG [26]; RARR [38]; Llatrieval [54]; QLM [55]
Pre-generation prompt engineering
UPRISE [56]
Context Inconsistency
Pre-generation prompt engineering
QUIP [57]; RECITE [58]; Smartbook [14]; PKG [59]; In-Context RALM [51]
Post-generation response correction
SourceCheckup [60]; EFEC [61]; RSEQGA [62]; WebCiteS [63]
Reasoning Deficiency
Pre-generation prompt engineering
SubgraphRAG [64]; Self-Reasoning [65]
Post-generation response correction
CoVe [66]; Searchain [67]; Verify and Edit [68]

Fig. 2. Taxonomy of Attribution for Mitigating six types of LLM Hallucinations.

and reorder them accordingly. C-RAG [25] evaluates reasoning paths and reranks documents based on verifiability rather than similarity.

Other approaches enhance attribution through more granular evidence. CoTAR [27] guides step-by-step reasoning to link reference segments to generated content. PRCA [28] condenses supportive sentences into concise generator inputs. RECOMP [29] summarizes references into query-relevant evidence. TOC [30] generates clarifying sub-questions, organizes them hierarchically, and validates responses to produce fully supported long-form outputs.

T4 Unverifiability may also occur when references are relevant but incomplete. Post-generation correction methods either (i) refine a single response using higher-quality evidence [37]–[42] or (ii) generate multiple responses and rank them by verifiability [31]–[34], [36]. Many techniques use an LLM or auxiliary verifier to assess reference support. Self-Refine [37] critiques and revises outputs while retrieving additional evidence. CaLM [40] verifies cited statements and retains only validated portions. AGREE [39] applies NLI to detect unsupported claims and trigger targeted retrieval. RARR [38] generates sub-questions to gather more evidence, and ITER-RETGEN [42] and ITRG [41] iteratively expand queries to refine responses.

A second family of methods produces multiple candidate responses and ranks them. TWEAK [34] measures supportiveness using a Hypothesis Verification Model. RLHF-based methods such as GopherCite [33] and Fine-Grained RLHF [36] score responses for verifiability and provide corrective feedback. LaMDA [31] filters outputs for factuality, while Sparrow [32] applies rule-based evaluation to re-rank or remove unsupported responses.

D. *Correcting Instruction Deviation*

Instruction deviation hallucination occurs when an LLM fails to follow instructions, leading to topic drift, incomplete answers, or inconsistencies. Attribution-based methods address this by using T1, T2, and T3.

T1 Instruction deviation often arises from ambiguous or underspecified queries. Query-refinement methods clarify user intent to ensure retrieval matches the task. Approaches include (i) keyword-based refinement [43], [44], (ii) query decomposition [30], [45]–[48], [53], and (iii) rule-based strategies [49]–[53]. MixAlign [43] identifies vague constraints and prompts users for clarification, while 1-PAGER [44] extracts keywords to iteratively filter irrelevant references. Blueprint [46] and related methods (CCV, DSP, TOC) decompose complex queries into sub-questions for targeted retrieval. Step-Back Prompting [48] broadens questions to improve reasoning.

TABLE I
TAXONOMY OF HALLUCINATION TYPES, DEFINITIONS, ORIGIN (RETRIEVER OR GENERATOR), EXAMPLES, AND MITIGATION TECHNIQUES.

Type	Definition	Ret.	Gen.	Example	Tech.
Overconfidence	Presenting uncertain or nuanced information as absolute fact.		✓	Query: How to lose weight? → Refs: [1] exercise; [2] diet, metabolism. Resp: The only way to lose weight is exercise.	T3, T4
Outdatedness	A response that was once correct but is now obsolete.	✓		Query: Who is the current U.S. president? → Ref: Donald Trump (pre-2025). Resp: Joe Biden is president.	T1, T2
Unverifiability	A response unsupported by any available evidence.	✓	✓	Query: Biological consequence of building Eiffel Tower? → Ref: No relevant records. Resp: It caused the extinction of the Parisian tiger.	T2, T4
Instruction Deviation	Output does not follow explicit user instructions.	✓	✓	Query: Translate "The weather is nice today." → Ref: Meaning is pleasant weather. Resp: Explains meaning instead of providing translation.	T1, T2, T3
Context Inconsistency	Response contradicts or ignores retrieved references.		✓	Query: Where is the Nile's source? → Ref: Located in central Africa. Resp: It starts in Egypt.	T3, T4
Reasoning Deficiency	Logical errors, invalid inference, or flawed reasoning chains.		✓	Query: Why do whales surface? → Ref: Whales breathe air via lungs. Resp: Whales use gills and do not need to surface.	T3, T4

Rule-based methods such as In-Context RALM [51], VTG [49], Query2doc [50], HyDE [52], and RRR [53] adjust or expand queries using contextual or semantic cues to better capture user intent.

T2 Instruction deviation may also occur when references are topically relevant but misaligned with user intent. Post-retrieval identification methods address this by re-ranking references based on semantic alignment [26], [38], [54], [55]. Self-RAG [26] combines dense similarity and keyword matching (TF–IDF, BM25) to prioritize relevant evidence. RARR [38] adds an NLI-based model for support-based reranking. LLatrieval [54] and QLM [55] use LLMs to iteratively check relevance and ensure the final set meets an alignment threshold.

T3 Pre-generation Prompt Engineering reduces instruction deviation by using prompt templates that explicitly encode the task before generation [56]. UPRISE [56] constructs a pool of templates from retrieved references and selects the most aligned one using an additional retriever, outperforming static template designs.

E. Aligning Context Inconsistency

Context-inconsistency hallucination occurs when an LLM contradicts retrieved references, often because parts of the response are not properly grounded. Attribution-based methods address this by using T3 and T4.

T3 Context inconsistency arises when references are incorrectly linked to response segments. Prompt engineering guides the model to stay consistent with retrieved evidence [14], [51], [57]–[59]. In-Context RALM [51] improves alignment by concatenating queries with retrieved documents. QUIP [57] explicitly prompts claim attribution. RECITE [58] has the model "recite" key information before generating an answer. SmartBook [14] uses sub-questions and keywords for explicit citation, while PKG [59] enriches prompts with domain-relevant background information to improve coherence and consistency.

T4 Even with correct references, the generator may produce misaligned content, leading to context-inconsistent hallucinations. Post-generation correction addresses this by comparing responses with retrieved evidence using either an auxiliary model [60], [61], [63] or the same LLM [62]. Most approaches rely on NLI-based verification: SourceCheckup and WebCiteS apply statement-level NLI checks to flag inconsistencies. EFEC identifies key tokens with an auxiliary LLM, masks them, and instructs the model to regenerate content based on retrieved evidence. RSEGQA, in contrast, prompts the same LLM to verify and revise its output by decomposing it into sub-statements, evaluating each against supporting references, and editing contradictions to maintain consistency.

F. Reducing Reasoning Deficiency

Reasoning deficiency hallucination often appears in Chain-of-Thought (CoT) settings, where flawed or incomplete reasoning leads to incorrect conclusions. Attribution-based methods mitigate this by using T3 and T4.

T3 Reasoning deficiencies can be reduced through structured prompts that guide step-by-step reasoning [64], [65]. SubgraphRAG [64] encodes references into a knowledge graph, extracts relevant subgraphs, and formats them into prompts supporting multi-step, evidence-grounded explanations. Self-Reasoning [65] builds prompts for each reasoning step, prompting the LLM to validate document relevance, extract key facts, and construct a coherent reasoning trajectory.

T4 Post-generation correction methods revise reasoning chains after generation, encouraging models to break responses into explicit steps, assess coherence, and repair flawed logic [66]–[68]. SearChain [67] uses predefined prompts to detect and regenerate faulty reasoning steps. Verify-and-Edit [68] measures step-level confidence, flags uncertainties, and generates retrieval-oriented questions for correction. CoVe [66] issues verification queries against retrieved references to ensure only validated reasoning steps are included in the final response.

IV. HALLUCINATION HANDLING USING ATTRIBUTION-BASED TECHNIQUES

Attribution-based methods mitigate hallucinations by identifying their source—either the *retriever* or the *generator* (Table I). Retriever-related issues arise from outdated, irrelevant, or incomplete references, while generator-related hallucinations occur during response formulation due to overconfidence or flawed reasoning. Retriever-oriented techniques (T1, T2) improve evidence quality: T1 reformulates queries by decomposing complex questions, clarifying ambiguity, or adding temporal cues to retrieve more accurate references, while T2 filters or re-ranks documents using semantic models or updated knowledge sources to ensure only relevant evidence is passed to the generator. Generator-oriented techniques (T3, T4) address hallucinations during or after generation: T3 reduces reasoning errors through structured prompts and task-specific keywords, and T4 verifies and revises outputs or selects the most consistent response from multiple candidates. These approaches leverage self-critique prompts, NLI-based validators, reward models, and confidence-ranking strategies to improve response accuracy and verifiability.

V. Discussion of Key Challenges

Despite their effectiveness, attribution-based methods face several challenges. Many rely on single-source validation, limiting support for open-domain or multi-perspective queries and potentially reinforcing bias. Dependence on external sources introduces risks related to incompleteness, reliability, and copyright. Using LLMs as retrieval judges (e.g., LLatrieval, CoVe) creates circularity, as hallucination-prone models evaluate one another. Long-context limitations persist for complex inputs and multiple subqueries, while chain-of-thought reasoning can amplify early errors without reliable checks. Prompt fragility also remains a concern, as template-based prompts lack generalization and few-shot prompting increases cost while still struggling with novel reasoning patterns.

VI. Conclusion

This survey examines attribution-based techniques for reducing hallucinations in RAG systems. We introduce a unified pipeline consisting of query refinement, reference identification, prompt engineering, and response correction, and map common hallucination types to their corresponding mitigation strategies. These methods enhance grounding, attribution, and factual consistency in LLM-generated responses.

Acknowledgment

This research is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore and Infocomm Media Development Authority.

References

[1] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen *et al.*, "Siren's song in the ai ocean: a survey on hallucination in large language models," *arXiv preprint arXiv:2309.01219*, 2023.

[2] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin *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, pp. 1–55, 2025.

[3] H. Ye, T. Liu, A. Zhang, W. Hua, and W. Jia, "Cognitive mirage: A review of hallucinations in large language models," *arXiv preprint arXiv:2309.06794*, 2023.

[4] S. Tonmoy, S. Zaman, V. Jain, A. Rani, V. Rawte, A. Chadha, and A. Das, "A comprehensive survey of hallucination mitigation techniques in large language models," *arXiv preprint arXiv:2401.01313*, vol. 6, 2024.

[5] D. Li, Z. Sun, X. Hu, Z. Liu, Z. Chen, B. Hu, A. Wu, and M. Zhang, "A survey of large language models attribution," *arXiv preprint arXiv:2311.03731*, 2023.

[6] S. Gupta, R. Ranjan, and S. N. Singh, "A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions," *arXiv preprint arXiv:2410.12837*, 2024.

[7] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, vol. 2, 2023.

[8] A. Azaria and T. Mitchell, "The internal state of an llm knows when it's lying," *arXiv preprint arXiv:2304.13734*, 2023.

[9] D. Lee, T. Whang, C. Lee, and H. Lim, "Towards reliable and fluent large language models: Incorporating feedback learning loops in qa systems," *arXiv preprint arXiv:2309.06384*, 2023.

[10] P. Manakul, A. Liusie, and M. J. Gales, "Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models," *arXiv preprint arXiv:2303.08896*, 2023.

[11] Y. Liang, Z. Song, H. Wang, and J. Zhang, "Learning to trust your feelings: Leveraging self-awareness in llms for hallucination mitigation," *arXiv preprint arXiv:2401.15449*, 2024.

[12] Z. Ji, L. Yu, Y. Koishekenov, Y. Bang, A. Hartshorn, A. Schelten, C. Zhang, P. Fung, and N. Cancedda, "Calibrating verbal uncertainty as a linear feature to reduce hallucinations," *arXiv preprint arXiv:2503.14477*, 2025.

[13] Y. Qin, Z. Cai, D. Jin, L. Yan, S. Liang, K. Zhu, Y. Lin, X. Han, N. Ding, H. Wang *et al.*, "Webcpm: Interactive web search for chinese long-form question answering," *arXiv preprint arXiv:2305.06849*, 2023.

[14] R. G. Reddy, D. Lee, Y. R. Fung, K. D. Nguyen, Q. Zeng, M. Li, Z. Wang, C. Voss, and H. Ji, "Smartbook: Ai-assisted situation report generation for intelligence analysts," *arXiv preprint arXiv:2303.14337*, 2023.

[15] B. Peng, M. Galley, P. He, H. Cheng, Y. Xie, Y. Hu, Q. Huang, L. Liden, Z. Yu, W. Chen *et al.*, "Check your facts and try again: Improving large language models with external knowledge and automated feedback," *arXiv preprint arXiv:2302.12813*, 2023.

[16] V. Karpukhin, B. Oguz, S. Min, P. S. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering." in *EMNLP (1)*, 2020, pp. 6769–6781.

[17] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders *et al.*, "Webgpt: Browser-assisted question-answering with human feedback," *arXiv preprint arXiv:2112.09332*, 2021.

[18] H. Qian, Y. Zhu, Z. Dou, H. Gu, X. Zhang, Z. Liu, R. Lai, Z. Cao, J.-Y. Nie, and J.-R. Wen, "Webbrain: Learning to generate factually correct articles for queries by grounding on large web corpus," *arXiv preprint arXiv:2304.04358*, 2023.

[19] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[20] Y. Zhang, S. Li, C. Qian, J. Liu, P. Yu, C. Han, Y. R. Fung, K. McKeown, C. Zhai, M. Li *et al.*, "The law of knowledge overshadowing: Towards understanding, predicting, and preventing llm hallucination," *arXiv preprint arXiv:2502.16143*, 2025.

[21] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *International conference on machine learning*. PMLR, 2020, pp. 3929–3938.

[22] M. J. Zhang and E. Choi, "Mitigating temporal misalignment by discarding outdated facts," *arXiv preprint arXiv:2305.14824*, 2023.

[23] W. Shi, S. Min, M. Yasunaga, M. Seo, R. James, M. Lewis, L. Zettlemoyer, and W.-t. Yih, "Replug: Retrieval-augmented black-box language models," *arXiv preprint arXiv:2301.12652*, 2023.

[24] Z. Yu, C. Xiong, S. Yu, and Z. Liu, "Augmentation-adapted retriever improves generalization of language models as generic plug-in," *arXiv preprint arXiv:2305.17331*, 2023.

[25] M. Kang, N. M. Gürel, N. Yu, D. Song, and B. Li, "C-rag: Certified generation risks for retrieval-augmented language models," 2024. [Online]. Available: https://arxiv.org/abs/2402.03181

[26] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, “Self-rag: Learning to retrieve, generate, and critique through self-reflection,” in *The Twelfth International Conference on Learning Representations*, 2023.

[27] M. Berchansky, D. Fleischer, M. Wasserblat, and P. Izsak, “Cotar: Chain-of-thought attribution reasoning with multi-level granularity,” *arXiv preprint arXiv:2404.10513*, 2024.

[28] H. Yang, Z. Li, Y. Zhang, J. Wang, N. Cheng, M. Li, and J. Xiao, “Prca: Fitting black-box large language models for retrieval question answering via pluggable reward-driven contextual adapter,” *arXiv preprint arXiv:2310.18347*, 2023.

[29] F. Xu, W. Shi, and E. Choi, “Recomp: Improving retrieval-augmented lms with compression and selective augmentation,” *arXiv preprint arXiv:2310.04408*, 2023.

[30] G. Kim, S. Kim, B. Jeon, J. Park, and J. Kang, “Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 996–1009.

[31] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du *et al.*, “Lamda: Language models for dialog applications,” *arXiv preprint arXiv:2201.08239*, 2022.

[32] A. Glaese, N. McAleese, M. Trebacz, J. Aslanides, V. Firoiu, T. Ewalds, M. Rauh, L. Weidinger, M. Chadwick, P. Thacker *et al.*, “Improving alignment of dialogue agents via targeted human judgements,” *arXiv preprint arXiv:2209.14375*, 2022.

[33] J. Menick, M. Trebacz, V. Mikulik, J. Aslanides, F. Song, M. Chadwick, M. Glaese, S. Young, L. Campbell-Gillingham, G. Irving *et al.*, “Teaching language models to support answers with verified quotes, 2022,” *URL https://arxiv. org/abs/2203.11147*, 2022.

[34] Y. Qiu, V. Embar, S. B. Cohen, and B. Han, “Think while you write: Hypothesis verification promotes faithful knowledge-to-text generation,” *arXiv preprint arXiv:2311.09467*, 2023.

[35] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave, “Atlas: Few-shot learning with retrieval augmented language models,” *Journal of Machine Learning Research*, vol. 24, no. 251, pp. 1–43, 2023.

[36] Z. Wu, Y. Hu, W. Shi, N. Dziri, A. Suhr, P. Ammanabrolu, N. A. Smith, M. Ostendorf, and H. Hajishirzi, “Fine-grained human feedback gives better rewards for language model training,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 59 008–59 033, 2023.

[37] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang *et al.*, “Self-refine: Iterative refinement with self-feedback,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 46 534–46 594, 2023.

[38] L. Gao, Z. Dai, P. Pasupat, A. Chen, A. T. Chaganty, Y. Fan, V. Y. Zhao, N. Lao, H. Lee, D.-C. Juan *et al.*, “Rarr: Researching and revising what language models say, using language models,” *arXiv preprint arXiv:2210.08726*, 2022.

[39] X. Ye, R. Sun, S. Ö. Arik, and T. Pfister, “Effective large language model adaptation for improved grounding and citation generation,” *arXiv preprint arXiv:2311.09533*, 2023.

[40] I. Hsu, Z. Wang, L. T. Le, L. Miculicich, N. Peng, C.-Y. Lee, T. Pfister *et al.*, “Calm: Contrasting large and small language models to verify grounded generation,” *arXiv preprint arXiv:2406.05365*, 2024.

[41] Z. Feng, X. Feng, D. Zhao, M. Yang, and B. Qin, “Retrieval-generation synergy augmented large language models,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 11 661–11 665.

[42] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen, “Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy,” *arXiv preprint arXiv:2305.15294*, 2023.

[43] S. Zhang, L. Pan, J. Zhao, and W. Y. Wang, “The knowledge alignment problem: Bridging human and external knowledge for large language models,” *arXiv preprint arXiv:2305.13669*, 2023.

[44] P. Jain, L. B. Soares, and T. Kwiatkowski, “1-pager: One pass answer generation and evidence retrieval,” *arXiv preprint arXiv:2310.16568*, 2023.

[45] J. Chen, G. Kim, A. Sriram, G. Durrett, and E. Choi, “Complex claim verification with evidence retrieved in the wild,” *arXiv preprint arXiv:2305.11859*, 2023.

[46] C. Fierro, R. K. Amplayo, F. Huot, N. De Cao, J. Maynez, S. Narayan, and M. Lapata, “Learning to plan and generate text with citations,” *arXiv preprint arXiv:2404.03381*, 2024.

[47] O. Khattab, K. Santhanam, X. L. Li, D. Hall, P. Liang, C. Potts, and M. Zaharia, “Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp,” *arXiv preprint arXiv:2212.14024*, 2022.

[48] H. S. Zheng, S. Mishra, X. Chen, H.-T. Cheng, E. H. Chi, Q. V. Le, and D. Zhou, “Take a step back: Evoking reasoning via abstraction in large language models,” *arXiv preprint arXiv:2310.06117*, 2023.

[49] H. Sun, H. Cai, B. Wang, Y. Hou, X. Wei, S. Wang, Y. Zhang, and D. Yin, “Towards verifiable text generation with evolving memory and self-reflection,” *arXiv preprint arXiv:2312.09075*, 2023.

[50] L. Wang, N. Yang, and F. Wei, “Query2doc: Query expansion with large language models,” *arXiv preprint arXiv:2303.07678*, 2023.

[51] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham, “In-context retrieval-augmented language models,” *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1316–1331, 2023.

[52] L. Gao, X. Ma, J. Lin, and J. Callan, “Precise zero-shot dense retrieval without relevance labels,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 1762–1777.

[53] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, “Query rewriting in retrieval-augmented large language models,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 5303–5315.

[54] X. Li, C. Zhu, L. Li, Z. Yin, T. Sun, and X. Qiu, “Llatrieval: Llm-verified retrieval for verifiable generation,” *arXiv preprint arXiv:2311.07838*, 2023.

[55] S. Zhuang, B. Liu, B. Koopman, and G. Zuccon, “Open-source large language models are strong zero-shot query likelihood models for document ranking,” *arXiv preprint arXiv:2310.13243*, 2023.

[56] D. Cheng, S. Huang, J. Bi, Y. Zhan, J. Liu, Y. Wang, H. Sun, F. Wei, D. Deng, and Q. Zhang, “Uprise: Universal prompt retrieval for improving zero-shot evaluation,” *arXiv preprint arXiv:2303.08518*, 2023.

[57] O. Weller, M. Marone, N. Weir, D. Lawrie, D. Khashabi, and B. Van Durme, “” according to...”: Prompting language models improves quoting from pre-training data,” *arXiv preprint arXiv:2305.13252*, 2023.

[58] Z. Sun, X. Wang, Y. Tay, Y. Yang, and D. Zhou, “Recitation-augmented language models,” *arXiv preprint arXiv:2210.01296*, 2022.

[59] Z. Luo, C. Xu, P. Zhao, X. Geng, C. Tao, J. Ma, Q. Lin, and D. Jiang, “Augmented large language models with parametric knowledge guiding,” *arXiv preprint arXiv:2305.04757*, 2023.

[60] K. Wu, E. Wu, A. Cassasola, A. Zhang, K. Wei, T. Nguyen, S. Riantawan, P. S. Riantawan, D. E. Ho, and J. Zou, “How well do llms cite relevant medical references? an evaluation framework and analyses,” *arXiv preprint arXiv:2402.02008*, 2024.

[61] J. Thorne and A. Vlachos, “Evidence-based factual error correction,” *arXiv preprint arXiv:2012.15788*, 2020.

[62] S. Huo, N. Arabzadeh, and C. Clarke, “Retrieving supporting evidence for generative question answering,” in *Proceedings of the annual international acm sigir conference on research and development in information retrieval in the Asia Pacific region*, 2023, pp. 11–20.

[63] H. Deng, C. Wang, X. Li, D. Yuan, J. Zhan, T. Zhou, J. Ma, J. Gao, and R. Xu, “Webcites: Attributed query-focused summarization on chinese web search results with citations,” *arXiv preprint arXiv:2403.01774*, 2024.

[64] M. Li, S. Miao, and P. Li, “Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation,” *arXiv preprint arXiv:2410.20724*, 2024.

[65] Y. Xia, J. Zhou, Z. Shi, J. Chen, and H. Huang, “Improving retrieval augmented language model with self-reasoning,” *arXiv preprint arXiv:2407.19813*, 2024.

[66] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston, “Chain-of-verification reduces hallucination in large language models,” *arXiv preprint arXiv:2309.11495*, 2023.

[67] S. Xu, L. Pang, H. Shen, X. Cheng, and T.-S. Chua, “Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks,” in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 1362–1373.

[68] R. Zhao, X. Li, S. Joty, C. Qin, and L. Bing, “Verify-and-edit: A knowledge-enhanced chain-of-thought framework,” *arXiv preprint arXiv:2305.03268*, 2023.