AI

MDPI

Systematic Review

From Illusion to Insight: A Taxonomic Survey of Hallucination Mitigation Techniques in LLMs

Ioannis Kazlaris *, Efstathios Antoniou, Konstantinos Diamantaras and Charalampos Bratsas

Department of Information and Electronic Engineering, International Hellenic University (IHU), 57400 Thessaloniki, Greece; antoniou@ihu.gr (E.A.); kdiamant@ihu.gr (K.D.); cbratsas@ihu.gr (C.B.)
* Correspondence: ikazlaris@ihu.gr

Abstract

Large Language Models (LLMs) exhibit remarkable generative capabilities but remain vulnerable to hallucinations—outputs that are fluent yet inaccurate, ungrounded, or inconsistent with source material. To address the lack of methodologically grounded surveys, this paper introduces a novel method-oriented taxonomy of hallucination mitigation strategies in text-based LLMs. The taxonomy organizes over 300 studies into six principled categories: Training and Learning Approaches, Architectural Modifications, Input/Prompt Optimization, Post-Generation Quality Control, Interpretability and Diagnostic Methods, and Agent-Based Orchestration. Beyond mapping the field, we identify persistent challenges such as the absence of standardized evaluation benchmarks, attribution difficulties in multi-method systems, and the fragility of retrieval-based methods when sources are noisy or outdated. We also highlight emerging directions, including knowledge-grounded fine-tuning and hybrid retrieval–generation pipelines integrated with self-reflective reasoning agents. This taxonomy provides a methodological framework for advancing reliable, context-sensitive LLM deployment in high-stakes domains such as healthcare, law, and defense.

Keywords: mitigation; hallucinations; large language models (LLMs); taxonomy

check for updates

Academic Editor: Isidoros Perikos

Received: 25 August 2025
Revised: 26 September 2025
Accepted: 30 September 2025
Published: 3 October 2025

Citation: Kazlaris, I.; Antoniou, E.; Diamantaras, K.; Bratsas, C. From Illusion to Insight: A Taxonomic Survey of Hallucination Mitigation Techniques in LLMs. *AI* **2025**, *6*, 260. https://doi.org/10.3390/ai6100260

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In this paper, we examine hallucinations in Large Language Models (LLMs), defined as instances where generated content appears coherent and plausible but contains factual inaccuracies or unverifiable claims [1–3]. These manifestations range from fabricated citations and logical inconsistencies to erroneous statistics and invented biographical details [4,5]. Such errors pose significant challenges for the use of LLMs in domains where factual accuracy is critical, including healthcare [6,7], law [8–10], and defense [11]. Some of the underlying causes include, but are not limited to, noisy training data, underrepresentation of minority viewpoints, outdated information, and the next-token log-likelihood objective that prioritizes plausible continuation over factual accuracy. Research has shown that hallucinations arise from the complex interplay between factual accuracy and generative capability [12–14]. Despite being mathematically inevitable, a plethora of mitigation strategies have been developed across pre-generation, generation, and post-generation phases. Pre-generation techniques include fine-tuning with human feedback [15] and contrastive learning [16] (distinct from contrastive decoding [17], which is applied during text generation). Architectural enhancements, by contrast, often incorporate decoding mechanisms [17], retrieval-based modules [18] and memory augmentations [19] to ground content in verifiable sources. During generation, structured prompting techniques such as

Chain of Thought (CoT) [20], guide reasoning toward evidence-based conclusions while post-generation safeguards employ self-consistency checks [21], self-verification and fact-checking systems [22], and self-verification loops [21,23], along with human-in-the-loop evaluations [24] to detect and rectify factual inconsistencies. In addition, internal model probing techniques and lightweight classifiers detect hallucination patterns through latent signals and linguistic cues [25,26]. AI agentic frameworks where a number of agents consult external tools, negotiate and self-reflect on their decisions complete our taxonomy [27,28]. We have structured our taxonomy into six categories: Training and Learning Approaches, Architectural Modifications, Input/Prompt Optimization, Post-Generation Quality Control, Interpretability and Diagnostic Methods, and Agent-Based Orchestration. A detailed account of each category follows in Section 4.2.

We organize the remainder of this paper as follows: Section 2 defines and categorizes hallucinations, discusses their underlying causes, and provides a brief overview of existing mitigation methods. Section 3 reviews related research, while Section 4 presents our methodology, proposed taxonomy and the key contributions. Section 5 provides an analytical discussion of mitigation strategies and Section 6 outlines benchmarks for evaluating hallucinations in LLMs. Section 7 offers a brief outline for high-stakes applications. Finally, Section 9 discusses the challenges in addressing hallucinations, and Section 10 concludes the paper.

2. Understanding Hallucinations

2.1. Definition of Hallucinations

Hallucinations in LLMs refer to instances where the generated content appears grammatically correct and coherent, yet it is factually incorrect, irrelevant, ungrounded (i.e.,cannot be traced to reliable sources), or logically inconsistent [1–3]. Distinct from simple errors and biases, hallucinations often appear as false claims about real or imaginary entities (people, events, places, or facts). They may also involve fabricated citations and data, such as non-existent biographical details [29,30], or take the form of erroneous statistics, inaccurate numerical values, and conclusions that do not follow logically from their premises [2,31]. Defining hallucinations is challenging because their classification depends heavily on the task, the logical relation between input and output, and pragmatic factors such as ambiguity, vagueness, presupposition, or metaphor [32,33]. The presence of hallucinations may be attributed to a multi-faceted and complex interplay of various factors such as sub-optimal training data [34,35], pre-training and supervised fine-tuning issues [36–38], or the probabilistic nature of sequence generation [12,13,39] as will be further analyzed in Section 2.3. Despite significant progress in LLM development, hallucinatory outputs continue to raise serious concerns about the reliability and trustworthiness of machine-generated text. These risks are particularly acute in critical domains such as healthcare [7], law [8,10] and defense [11], where considerable harm may be caused to individuals.

However, hallucinations in LLMs may be viewed not merely as technical limitations but as fundamental mathematical inevitabilities inherent to their architecture and function. Training LLMs for predictive accuracy inevitably produces hallucinations, regardless of model design or data quality, due to the probabilistic nature of language generation and the inability of LLMs to learn all computable functions [12–14]. This limitation is further supported by computational learning theory constraints and the lack of complete factual mappings [12,13]. The resulting tension arises because the next-token prediction objective favors statistically likely continuations, prioritizing linguistic plausibility over epistemic truth.

Despite the paradox between factual accuracy and probabilistic capability, several studies challenge the view of hallucinations as universal limitations. They argue that

this inevitability is not uniformly distributed across tasks [40] and that hallucinations can function as exploitable features for adversarial robustness [41]. Research shows that intentional hallucination can generate out-of-distribution concepts, boosting performance in tasks such as poetry and storytelling by 24% through associative chain disruption [42]. However, if left unregulated, this process may lead to extreme confabulation. Layer-wise analyses further suggest that certain architectural layers in LLMs are critical points where hallucination and creativity can be balanced [43]. Similarly, in tasks such as hypothesis generation or brainstorming, the unlikely combinations of entities and reasoning processes produced by hallucinations may reveal insights that conventional methods would overlook. For instance, combinatorial entropy has been shown to generate viable hypotheses in physics and biology [44], while controlled hallucination has been applied to brainstorming through reflective prompting [45]. Given this dual nature of hallucinations, we believe that their evaluation needs to occur within the context of specific applications and under cultural sensitivity awareness, so as to distinguish between scenarios that demand factual rigor and those that benefit from creative exploration [46–48].

2.2. Categories of Hallucinations

We can categorize hallucinations in LLMs into distinct types based on their relationship to source material, model knowledge boundaries, and linguistic structure. Following the majority of researchers, we distinguish between intrinsic hallucinations (factuality errors) and extrinsic hallucinations (faithfulness errors) as foundational categories [2,3,31,49–53]. These are distinct from other phenomena such as bias, which some scholars classify as a hallucination subtype [2,3,54–56] while others approach as a separate representational skew not necessarily involving factual error [1,48,57]. Furthermore, we explicitly position intrinsic (factuality) and extrinsic (faithfulness) hallucinations as subcategories of the broader intrinsic/extrinsic taxonomy, thus aligning with research in [58] which emphasizes that factuality requires alignment with external truths, while faithfulness requires alignment with source material. Acknowledging the research of [59], where a task-independent categorization of hallucinations is presented (Factual Mirage and Silver Lining in addition to their subcategories), we summarize the major categories thus:

- Intrinsic hallucinations (factuality errors) occur when a model generates content that contradicts established facts, its training data, or referenced input [2,3,31,49,50,52,53]. Following the taxonomic names in [5] the subtypes of this category may include (but are not limited to):
    - Entity-error hallucinations, where the model generates non-existent entities or misrepresents their relationships (e.g., inventing fake individuals, non-existent biographical details [4] or non-existent research papers), often measured via entity-level consistency metrics [60], as shown in [3,31,57,61].
    - Relation-error hallucinations, where inconsistencies are of a temporal, causal, or quantitative nature, such as erroneous chronologies or unsupported statistical claims [2,31,57].
    - Outdatedness hallucinations, which are characterized by outdated or superseded information, often reflecting a mismatch between training data and current knowledge as shown in [1,3,51,62].
    - Overclaim hallucinations, where models exaggerate the scope or certainty of a claim, often asserting more than the evidence supports [5,49,56,57].
- Extrinsic hallucinations (faithfulness errors) appear when the generated content deviates from the provided input or user prompt. These hallucinations are generally characterized by the inability to verify the generated output which may or may not be

true but, in either case, it is either not directly deducible from the user prompt or it contradicts itself [2,3,5,50,56,58]. Extrinsic hallucinations may manifest as:

- Incompleteness hallucinations, which occur when answers are truncated or necessary context that is useful is omitted [2,51,57].
- Unverifiability hallucinations, where outputs that neither align with available evidence nor can be clearly refuted are present [5,56].
- Emergent hallucinations, defined as those arising unpredictably in larger models due to scaling effects [63]. These can be attributed to cross-domain reasoning and modality fusion especially in multi-modal settings or Chain of Thought (CoT) prompting scenarios [3,63,64], multi-step inference errors [65] and abstraction or alignment issues as shown in [50,57,64]. For instance, self-reflection demonstrates mitigation capabilities, effectively reducing hallucinations only in models above a certain threshold (e.g., 70B parameters), while paradoxically increasing errors in smaller models due to limited self-diagnostic capacity [5].

More importantly, certain types of hallucinations can compound over time in lengthy prompts or multi–step reasoning tasks, thus triggering a cascade of errors. This phenomenon, known as the snowball effect, further erodes user trust and model reliability [65–67].

2.3. Underlying Causes of Hallucinations

Hallucinations in LLMs stem from a variety of interwoven factors spanning the entire development pipeline. Although the impact of each cause may vary depending on the task and model architecture, we present them here in a non-hierarchical format to emphasize their interconnectedness. These factors include (but are not limited to) model architecture and decoding strategies [5,12–14,39], data quality, bias, memorization and poor alignment [15,34,35,37,39,68,69], pre-training and supervised fine-tuning issues [36–38], compute scale issues and under-training [5,38], prompt design and distribution shift [70–72], and retrieval-related mismatches [73,74].

In terms of model architecture, the use of decoding strategies and the inherently probabilistic nature of sequence-to-sequence modeling is a primary cause of hallucinations. For instance, the next-token log-likelihood objective commonly used during training which prioritizes plausible continuation over factual accuracy [12,13] has led to a significant body of research on decoding strategies such as nucleus sampling [75], contrastive decoding [17], and confidence-aware decoding [76] among others. Recent work also connects hallucinations to a lack of explicit modeling of uncertainty or factual confidence [77–79].

A model's vast and oftentimes noisy training data may often suffer from the inclusion of inaccuracies, contradictions (due to the presence of entirely or partially conflicting data also known as knowledge overshadowing), and biases, the presence of which may hinder the model's ability to generate factually reliable content [54,80]. While biases can cause hallucinations, and hallucinations represent a significant category of LLM output errors, in this systematic review we specifically examine the nature and mitigation of generated text that is factually ungrounded or inconsistent. If, for instance, a model is trained on imbalanced data, i.e., data skewed towards a majority perspective, favoring specific cultural differences or even personal beliefs, it might generate content that contradicts the realities or experiences of minority groups due to lack of exposure to their narratives [39,46,81]. These biases may be attributed to the underrepresentation of specific groups or ideas, making it challenging for models to generate content that is truly ethical and free from prejudice.

Data duplication and repetition, both of which may lead to overfitting and memorization issues, further degenerate the quality of the model output as shown in [34,35]. Ambiguous data may hinder the quantification or elicitation of model uncertainty, with

research suggesting that scaling, although generally beneficial, may have reached a fundamental limit, where models merely memorizing disambiguation patterns instead of reasoning [68]. Additionally, preference-fine-tuning achieved with Reinforcement Learning from Human Feedback (RLHF) can reward fluent but significantly polarized answers, a form of alignment-induced hallucination [38]. This kind of RL-based preference optimization has been linked to sycophantic agreement with user misconceptions further exacerbating factual inconsistencies [37] while recent work demonstrates that such hallucinations arise partly from reward hacking during alignment, where models prioritize reward signals over truthfulness and are addressed with constrained fine-tuning methods [61].

The high computational cost and prolonged training times required for training LLMs are additional factors that affect the quality of LLM generation: training data are not frequently updated thus hindering their ability to provide up-to-date and accurate information [37,38]. Controlled experiments using LLM behavioral probing have also demonstrated that verbatim memorization of individual statements at the sentence level as well as statistical patterns of usage at the corpora level (template-based reproduction) can also trigger hallucinations [35,36], while another line of research has demonstrated that many LLMs are under-trained, and that optimal performance requires scaling both model size and the number of training tokens proportionally [38].

In terms of prompting techniques, poor or insufficient prompting can significantly contribute to the generation of hallucinations since the omission of important information or failure to restrict the model's output can introduce ambiguity [82]. In many such cases, the model defaults to probabilistic pattern-matching which favors generalization over user intent as there is little or no information in the prompt that would enable it to deliver a more concise and targeted response [70–72]. Adversarial or out-of-distribution prompts can also have a detrimental effect as shown in [20,82] while in long-context models, inaccuracies may also emerge from summarization or retrieval failures during extended sequence attention, which can cause the model to lose track of grounding over time [67,83]. Furthermore, low-resource settings lack mitigation options requiring fine-tuning; zero-shot detection methods like self-consistency probes [84] or hesitation analysis [62] detect hallucinations without significant computational overhead.

While the above causes are presented in a non-hierarchical format to emphasize their interconnectedness, they can also be viewed as clustering around broader dimensions:

- data-related issues (e.g., noisy, biased, duplicated, or imbalanced training corpora),
- model and training-related issues (e.g., probabilistic decoding objectives, memorization, under-training, or alignment-induced effects), and
- usage-related issues (e.g., prompt design, distribution shift, and retrieval mismatches).

These categories are not mutually exclusive, but they provide a higher-level structure that connects naturally to the mitigation strategies reviewed later. For example, noisy training data is primarily addressed through Section 5.1 Training and Learning Approaches, architectural and decoding limitations through Section 5.2 Architectural Modifications, and prompt-related vulnerabilities through Section 5.3 Input/Prompt Optimization. We revisit these linkages in Section 4, where the proposed taxonomy explicitly maps mitigation strategies to the underlying causes they target.

3. Related Works

Research on hallucinations in LLMs has expanded rapidly, offering a diverse range of perspectives on their causes, forms, and mitigation strategies. Several surveys provide comprehensive overviews of hallucination mitigation across the LLM pipeline, from pre-generation to post-generation phases as seen in [1–3,31,49,59,85]. Much of the existing literature has adopted a task-oriented taxonomy, categorizing mitigation techniques

according to downstream applications (e.g., summarization, question answering, code generation) [86] or by the level of system intervention (e.g., pre-training, fine-tuning, or decoding) [2,31,56,59]. While this body of research offers valuable insights into where hallucinations occur and what tasks they affect, it often underrepresents the how—i.e., the core methodological principles underlying mitigation strategies. Some works have begun to address this by introducing method-oriented taxonomies which classify strategies based on techniques such as data augmentation, retrieval integration, prompt engineering or external fact-checking [3,37,52]. Notably, some of these studies focus on specific mitigation classes such as retrieval-augmented generation [51,87] ambiguity and uncertainty estimation [68], knowledge graphs [88], multi-agent reinforcement learning [27], thus offering deep insights into a narrow family of techniques. Others, like [3], begin to outline broad method-based categories but do not always explore fine-grained distinctions such as whether a method is self-supervised, classifier-guided, contrastive, or interpretability-driven.

As Figure 1 illustrates, our paper aims to complement this current body of research by offering an updated, method-oriented taxonomy that focuses exclusively on hallucination mitigation in textual LLMs. Our taxonomy is designed to clarify strategies based on their methodological foundation—such as training and learning approaches, architectural modifications, prompt engineering, output verification, post-processing, interpretability, and agent-based orchestration. By delving into the structural properties of existing mitigation strategies, we hope to clarify overlapping concepts and identify gaps that may serve as opportunities for future research.

Review Paper	Task-Oriented	Method-Oriented	Multimodal	Text	Fine-Grained
Tonmoy, 2024	✓	✗	✗	✓	✗
Yin et al., 2024	✓	✓	✗	✓	✗
Rawte et al., 2023	✓	✗	✗	✓	✗
Huang et al., 2024	✓	✓	✓	✗	✗
Li et al., 2025	✗	✓	✓	✗	✗
Zhao et al., 2025	✓	✓	✗	✓	✓
Agrawal et al., 2024	✗	✓	✗	✓	✓
Liu et al., 2024	✓	✓	✗	✗	✓
Luo et al., 2024	✓	✓	✗	✓	✗
Zhang & Zhang, 2025	✗	✓	✗	✓	✓
Perković et al.,2024	✗	✓	✗	✓	✗
Bilal et al., 2025	✗	✓	✗	✓	✓
Zhang et al., 2023	✓	✗	✗	✓	✗
Ji et al., 2024	✓	✗	✗	✓	✗
Rawte et al., 2023	✓	✗	✗	✓	✗
Lin et al., 2024	✓	✓	✗	✓	✗
Our Paper	✓	✓	✗	✓	✗

Figure 1. Comparison of hallucination survey taxonomies in LLM research, highlighting differences in categorization schemes such as task-oriented, system-level, and method-based classifications [1–3,27,31,37,49,51,52,56,59,68,85–88].

4. Review Methodology, Proposed Taxonomy, Contributions and Limitations

4.1. Review Methodology

Defining a single, unchanging taxonomy to classify and address hallucinations is a challenging task that can be attributed at least partially to the complex and interwoven

nature of the factors that cause them. In this paper, we try to address this challenge by presenting a 5-stage hierarchical framework:

- Literature Retrieval: We systematically collected research papers from major electronic archives—including Google Scholar, ACM Digital Library, IEEE Xplore, Elsevier, Springer, and ArXiv—with a cutoff date of 12 August 2025. Eligible records were restricted to peer-reviewed journal articles, conference papers, preprints under peer review, and technical reports, while non-academic sources such as blogs or opinion pieces were excluded. A structured query was used, combining keywords: ("mitigation" AND "hallucination" AND "large language models") OR "evaluation". In addition, we examined bibliographies of retrieved works to identify further relevant publications.
- Screening: The screening process followed a two-stage approach. First, titles and abstracts were screened for topical relevance. Records passing this stage underwent a full-text review to assess eligibility. Out of 412 initially retrieved records, 83 were excluded as irrelevant at the screening stage. The 329 eligible papers were then examined in detail and further categorized into support studies, literature reviews, datasets/benchmarks, and works directly proposing hallucination detection or mitigation methods. The final set of 221 studies formed the basis of our taxonomy. This process is summarized in Figure 2.

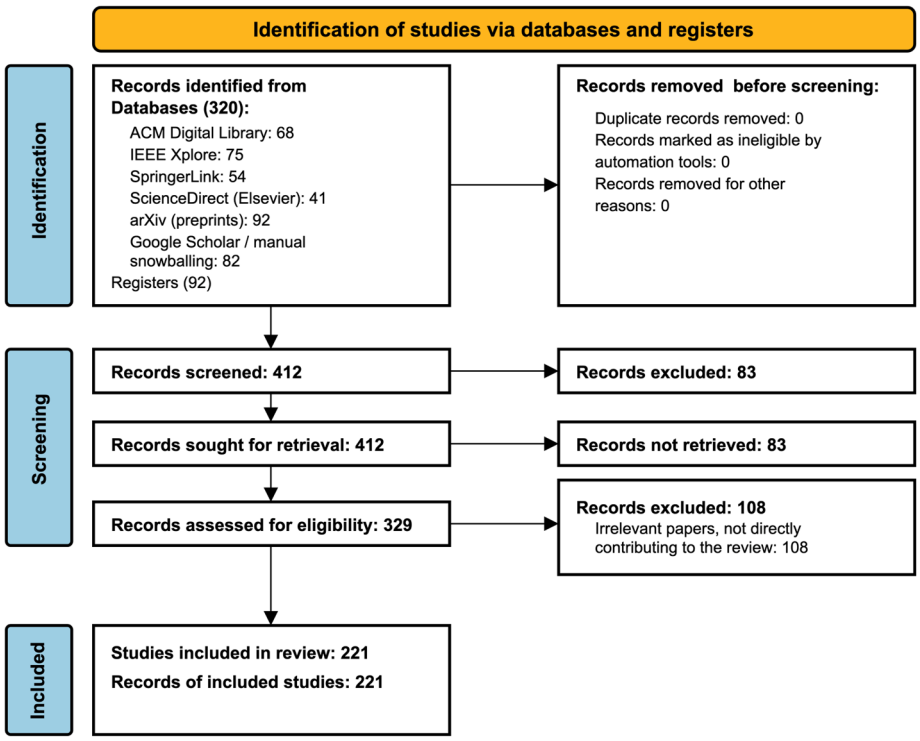

Figure 2. PRISMA flow diagram of study selection.

- Paper-level tagging, where every study was assigned one or more tags corresponding to its employed mitigation strategies. Our review accounts for papers that propose multiple methodologies by assigning them multiple tags, ensuring a comprehensive representation of each paper's contributions.
- Thematic clustering, where we consolidated those tags into six broad categories presented analytically in Section 4.2. This enabled us to generate informative visualizations that reflect the prevalence and trends among different mitigation techniques.

- Content-specific retrieval: To gain deeper insight into mitigation strategies, we developed a custom Retrieval-Augmented Generation (RAG) system based on the Mistral language model as an additional research tool, which enabled us to extract content-specific passages directly from the research papers.

This systematic review follows the PRISMA 2020 guidelines (see the Supplementary Materials) and has been registered with the Open Science Framework (OSF) under the registration ID: zbdy2. The registration details are publicly available at https://osf.io/zbdy2 (accessed 26 September 2025).

4.2. Proposed Taxonomy and Review Organization

Building upon the systematic methodology outlined in Section 4.1, our review directly addresses the challenges inherent in classifying diverse hallucination mitigation techniques. A key difficulty is that the majority of the papers implement a number of strategies (oftentimes referred to as frameworks), and therefore it is not always clearly discernible what the primary mitigation method is and to what extent its results can be safely attributed to it. We acknowledge this fact by explaining at the start of every major category our placement rationale. Our taxonomy organizes hallucination mitigation strategies into six main, method-focused categories, as shown in Figure 3.

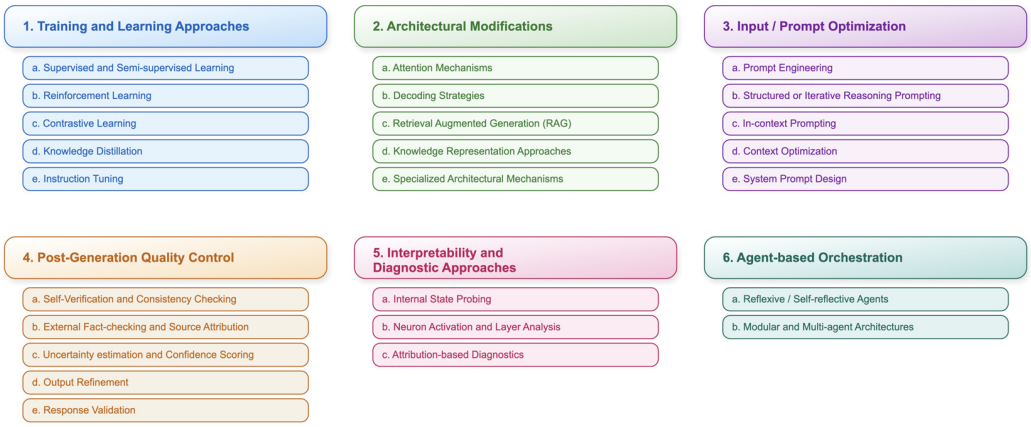

Figure 3. Visualization of the proposed taxonomy for hallucination mitigation, organizing strategies into six method-based categories.

1. Training and Learning Approaches (Section 5.1): Encompasses diverse methodologies employed to train and refine AI models, shaping their capabilities and performance.
2. Architectural Modifications (Section 5.2): Covers structural changes and enhancements made to AI models and their inference processes to improve performance, efficiency, and generation quality.
3. Input/Prompt Optimization (Section 5.3): Focuses on strategies for crafting and refining the text provided to AI models to steer their behavior and output, often specifically to mitigate hallucinations.
4. Post-Generation Quality Control (Section 5.4): Encompasses essential post-generation checks applied to text outputs, aiming to identify or correct inaccuracies.
5. Interpretability and Diagnostic Approaches (Section 5.5): Encompasses methods that help researchers understand why and where a model may be hallucinating (e.g., Internal State Probing, Attribution-based diagnostics).
6. Agent-based Orchestration (Section 5.6): Includes frameworks comprising single or multiple LLMs within multi-step loops, enabling iterative reasoning and tool usage.

In Section 5, we outline the six categories by summarizing their main principles. We then review research papers that apply methods in each category. To highlight specific details of these methods, we also include "support papers"—works that, while not directly focused on hallucination mitigation, provide relevant insights, evidence, or results related to the methods.

4.3. Contributions and Key Findings

To complement our taxonomy, we present three visualizations that synthesize key trends in hallucination mitigation research and may indicate potential future research directions.

Figure 4 shows the distribution of papers across mitigation techniques. Prompt Engineering dominates due to its low cost, flexibility, and ability to guide model behavior without retraining. Supervised/Semi-supervised Learning, External Fact-checking, and Output Refinement follow, highlighting interest in post hoc refinement and verification. Less explored areas, such as Self-reflective Agents and Knowledge Distillation, point to opportunities for further research.

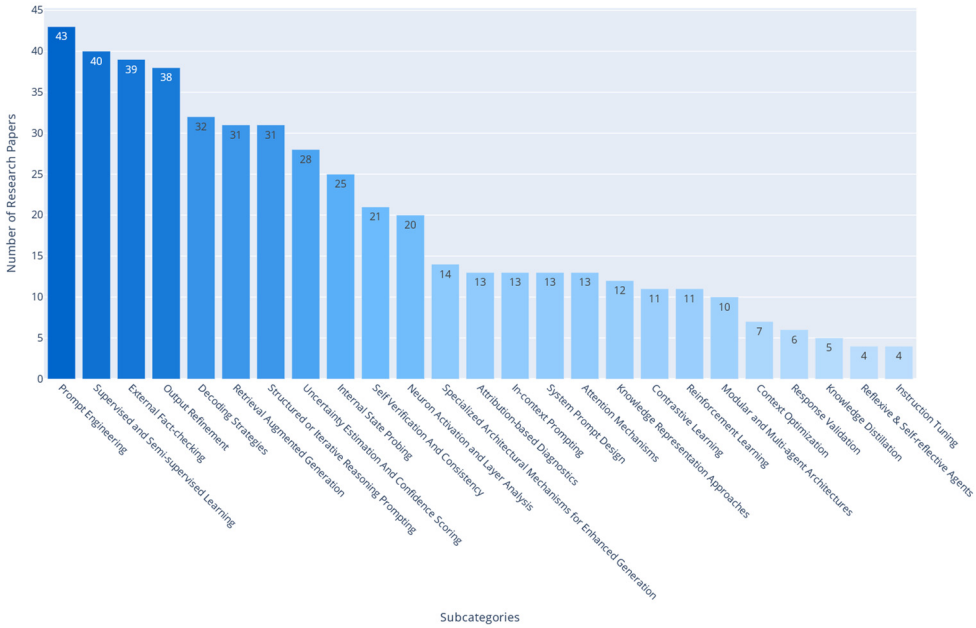

Figure 4. Distribution of research papers across individual hallucination mitigation subcategories, showing the prevalence of methods such as prompt engineering, external fact-checking, and decoding strategies.

Figure 5 traces the temporal growth of hallucination mitigation research. The trendline shows a noticeable surge in publications from 2022 onward, particularly following the public attention garnered by generative LLMs like ChatGPT. Notably, over 100 papers were published in 2023 alone, signaling both the urgency and the complexity of the hallucination problem as LLMs become integrated into high-stakes applications.

Figure 6 aggregates the number of papers per top-level category in our taxonomy, revealing that Input/Prompt Optimization, Architectural Modifications, and Training and Learning Approaches constitute the bulk of current research. This higher-level summary not only validates the comprehensiveness of the taxonomy but also provides researchers with a macro-level understanding of where scholarly attention is concentrated and where new contributions might be most impactful. Additionally, we have synthesized a comparison table of hallucination-mitigation subcategories derived from our literature review (Appendix A).

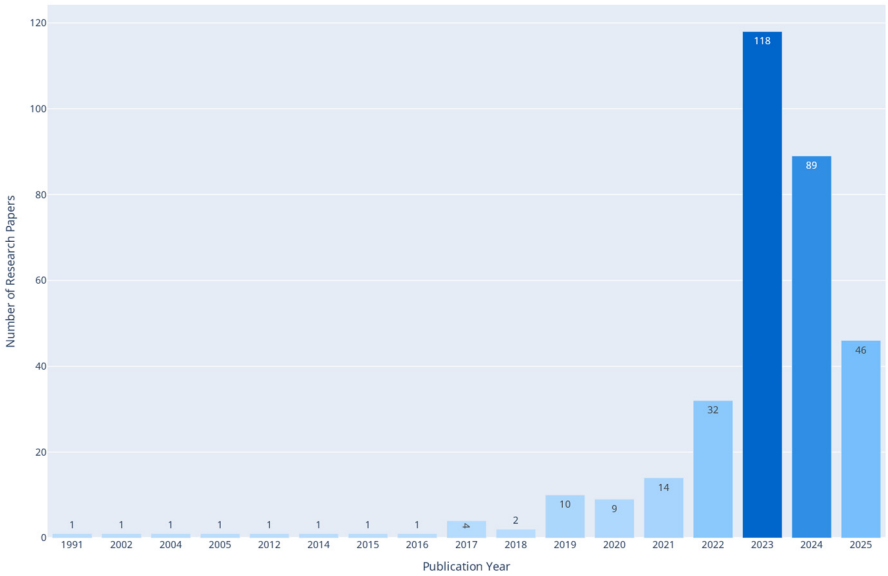

Figure 5. Yearly publication trend of hallucination mitigation research in LLMs, indicating a sharp increase in scholarly output since 2022.

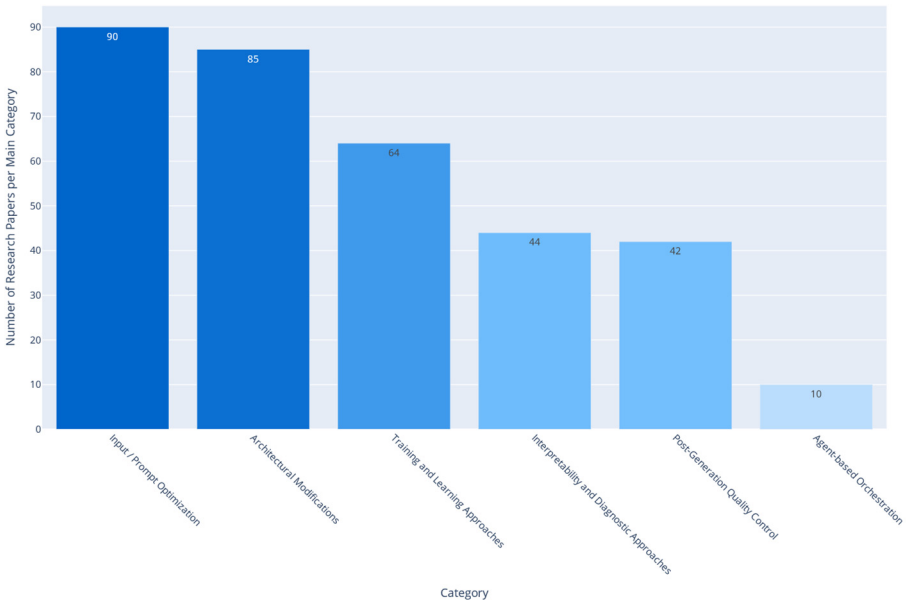

Figure 6. Aggregated number of research papers per top-level category in the proposed taxonomy.

Our taxonomy not only organizes existing literature but also reveals research gaps and emerging directions. By categorizing hallucinations by root causes, manifestations, and domain-specific effects, it highlights uneven scholarly attention. While areas like Prompt Engineering, Supervised/Semi-supervised Learning, and External Fact-checking dominate, others—such as Knowledge Distillation, Response Validation, and Self-reflective Agents—remain underexplored, offering opportunities for further work. The sharp rise in publications since 2022, underscores growing interest and methodological evolution.

5. Methods for Mitigating Hallucinations

5.1. Training and Learning Approaches

Training and Learning Approaches encompass a diverse set of methodologies employed to train and refine AI models. We have included foundational approaches like Supervised and Semi-supervised Learning, learning via interaction and reward signals in

Reinforcement Learning, and differentiating examples using Contrastive Learning. We have also included techniques such as Knowledge Distillation transferring knowledge from larger to smaller models, and instruction tuning, which aligns models to follow natural language instructions.

5.1.1. Supervised and Semi-Supervised Learning

Supervised learning [15,89] and semi-supervised learning [90,91] are foundational approaches that rely on high-quality datasets to ensure factual accuracy and effective generalization. Supervised learning uses annotated data to align model outputs with desired outcomes, while semi-supervised learning combines labeled and unlabeled data to reduce annotation costs. Their effectiveness depends on dataset quality, with well-curated and diverse examples helping to minimize bias and hallucinations [34–38]. Such datasets typically undergo iterative review and validation. Based on these methodologies, we group the papers in this subcategory as follows:

- Fine-Tuning with factuality objectives, where techniques such as FactPEGASUS make use of ranked factual summaries for factuality-aware fine-tuning [92] while FAVA generates synthetic training data using a pipeline involving error insertion and post-processing to address fine-grained hallucinations [93]. Faithful Finetuning applies weighted cross-entropy and fact-grounded QA losses to enhance faithfulness [94], while Principle Engraving fine-tunes LLaMA on self-aligned, principle-based responses [95]. Other work [5] examines the interplay between supervised fine-tuning and RLHF in mitigating hallucinations. Adversarial approaches build on Wasserstein GANs [90], with AFHN synthesizing features for new classes using labeled samples as context, supported by classification and anti-collapse regularizers to ensure feature discriminability and diversity.
- Synthetic Data and Weak Supervision, where studies automatically generated hallucinated data or weak labels for training. For instance, in [91] hallucinated tags are prepended to the model inputs so that it can learn from annotated examples to control hallucination levels while [96] uses BART and cross-lingual models with synthetic hallucinated datasets for token-level hallucination detection. Similarly, Petite Unsupervised Research and Revision (PURR) involves fine-tuning a compact model on synthetic data comprising corrupted claims and their denoised versions [97] while TrueTeacher uses labels generated by a teacher LLM to train a student model on factual consistency [98].
- Preference-Based Optimization and Alignment: In [99] a two-stage framework first combines supervised fine-tuning using curated legal QA data and Hard Sample-aware Iterative Direct Preference Optimization (HIPO) to ensure factuality by leveraging signals based on human preferences while in [80] a lightweight classifier is finetuned on contrastive pairs (hallucinated vs. non-hallucinated outputs). Similarly, mFACT—a metric for factual consistency—is derived from training classifiers in different target languages [100], while Contrastive Preference Optimization (CPO) combines a standard negative-log likelihood loss with a contrastive loss to finetune a model on a dataset consisting of triplets (source, hallucinated translation, corrected translation) [101]. UPRISE employs a retriever model that is trained using signals from an LLM to select optimal prompts for zero-shot tasks, allowing the retriever to directly internalize alignment signals from the LLM [102]. Finally, behavioral tuning uses label data (dialogue history, knowledge sources, and corresponding responses) to improve alignment [103].
- Knowledge-Enhanced Adaptation: Techniques like HALO injects Wikidata entity triplets or summaries via fine-tuning [104] while Joint Entity and Summary Generation

employs a pre-trained Longformer model which is finetuned on the PubMed dataset, in order to mitigate hallucinations by supervised adaptation and data filtering [105]. The impact of injecting new knowledge into LLMs via supervised finetuning and the potential risk of hallucinations is also studied in [106].

- Hallucination Detection Classifiers: [107] involves fine-tuning a LLaMA-2-7B model to classify hallucination-prone queries using labeled data while in [108] a sample selection strategy improves the efficiency of supervised fine-tuning by reducing annotation costs while preserving factuality through supervision.

Beyond their primary use, supervised and semi-supervised learning also serve as enabling methods that complement approaches such as Contrastive Learning, Internal State Probing, Retrieval-Augmented Generation, and Prompt Engineering. They have been applied to tasks including answer aggregation from annotated data [109], training factuality classifiers [83,110–112], generating synthetic datasets [113,114], and contributing to refinement pipelines [15,53,115–123]. More specifically:

- Training of factuality classifiers: Supervised finetuning is used to train models on labeled text data in datasets such as HDMBENCH, TruthfulQA, and multilingual datasets demonstrating improvements in task-specific performance and factual alignment [110,111,118]. Additionally, training enables classifiers to detect properties such as honesty and lies within intermediate representations resulting in increased accuracy and separability of these concepts as shown in [83,112,117].
- Synthetic data creation, which involves injecting hallucinations into correct reasoning steps, as in FG-PRM, which trains Process Reward Models to detect specific hallucination types [113]. Approaches like RAGTruth provide human-annotated labels on response grounding to support supervised training and evaluation [114], while [124] introduces an entity-substitution framework that generates conflicting QA instances to address over-reliance on parametric knowledge.
- Refinement pipelines employ supervised training in various forms, such as critic models trained on LLM data with synthetic negatives [115], augmentation techniques like TOPICPREFIX for improved grounding [116], and models such as HVM trained on the FATE dataset to distinguish faithful from unfaithful text [119]. Related methods target truthful vs. untruthful representations [53,120,121], while self-training on synthetic data outperforms crowdsourced alternatives [122]. Finally, WizardLM fine-tunes LLaMA on generated instructions, enhancing generalization [123].

5.1.2. Reinforcement Learning

Reinforcement Learning (RL) (Figure 7) trains models by optimizing reward signals rather than relying on labeled data, learning through trial-and-error to maximize cumulative rewards [125]. For example, NLI-based reward functions evaluate whether summaries are entailed by source documents, encouraging fluent and grounded outputs with fewer hallucinations [126]. Other approaches fine-tune LLMs with factuality-based signals such as FactScore and confidence-derived metrics [127]. Additionally, techniques like process supervision improve both generation and reward modeling by ranking multi-step reasoning chains, forming the basis for systems such as InstructGPT [128].

Defining precise and scalable reward functions in standard RL is challenging, leading to the development of Reinforcement Learning from Human Feedback (RLHF) [15,129]. In RLHF, human feedback guides model outputs to align with styles, constraints, or preferences, which helps reduce hallucinations by improving accuracy and encouraging models to acknowledge uncertainty [15,129]. The reward model provides the optimization signal, enhancing factuality, coherence, and ethical alignment [15,81,130,131]. Notable results include Anthropic's Constitutional AI, which achieved an 85% reduction in harmful

hallucinations [81], and OpenAI's GPT-4, which showed a 19% improvement over GPT-3.5 on adversarial factuality benchmarks and strong gains on TruthfulQA [131]. More recent work explores reward-model-guided fine-tuning, where self-evaluation signals push models to abstain when queries exceed their parametric knowledge, converting potential hallucinations into refusals [132].

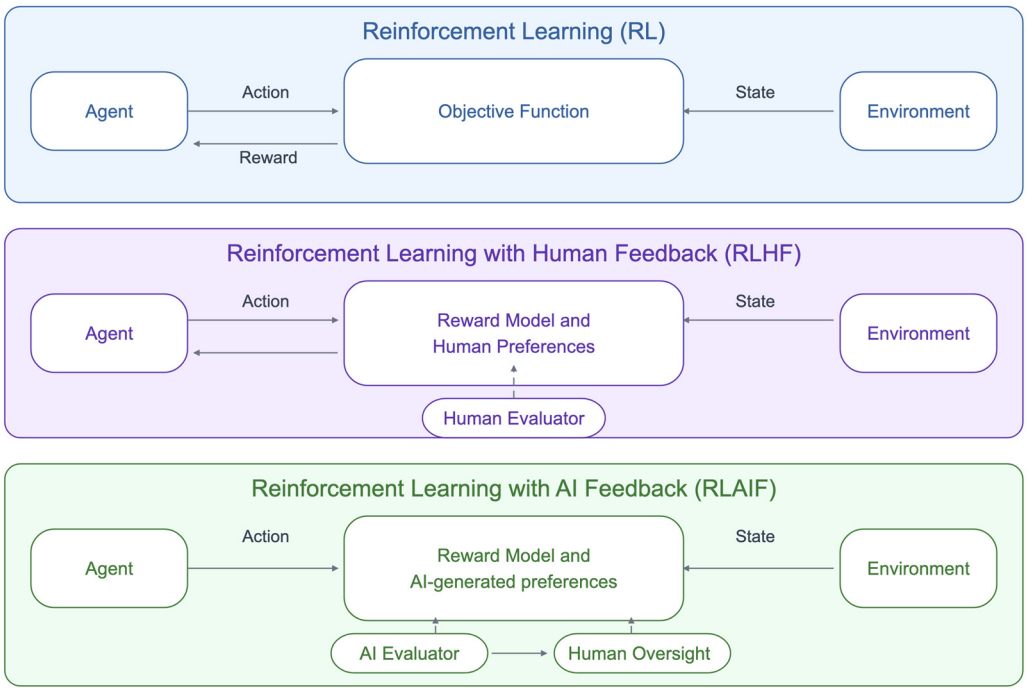

Figure 7. Overview of reinforcement learning for hallucination mitigation, including RLHF and RLAIF.

While RLHF has proven effective, its application is not without challenges. Research shows that extensive RL training can increase context-dependent sycophancy, with models favoring user agreement over factual accuracy [69]. Moreover, RLHF is resource-intensive, time-consuming, and subject to human and cultural biases, while inverse scaling issues may cause models to express stronger political views or resist certain instructions [37,81,133]. To address these limitations, Reinforcement Learning from AI Feedback (RLAIF) replaces or supplements human evaluators with AI systems, automating feedback generation and often surpassing RLHF in helpfulness, honesty, and harmlessness while dramatically reducing annotation costs [81,134]. For instance, RLAIF achieved a tenfold reduction in human labor, and models trained on GPT-4 critiques outperformed RLHF counterparts on MT-Bench and AlpacaEval [134]. Related approaches include Reinforcement Learning with Knowledge Feedback (RLKF), which uses factual preferences derived from the DreamCatcher tool to train reward models with PPO [135]. Similarly, HaluSearch frames response generation as a dual-process system that alternates between fast and slow reasoning, employing Monte Carlo Tree Search (MCTS) and self-evaluation reward signals to guide reasoning paths away from hallucinations through step-level verification [136].

RLHF and RLAIF fall within the broader paradigm of Alignment Learning [130], which aims to ensure AI systems act in line with human values and intentions [81,130,137,138]. Despite the costs of data collection, RL and its variants typically improve factual consistency and user satisfaction [15], while advanced forms such as hierarchical and multi-agent RL are being explored to refine alignment [139]. Reinforcement Learning has also been identified as a key enabler for multi-agent LLM systems, supporting meta-thinking abilities like self-reflection and adaptive planning, thereby fostering more trustworthy

agents [27,28]. While RL/RLHF/RLAIF can make use of AI agents that function as learners or decision-makers [15,125,131], we categorize RL, RLHF, and RLAIF under Training and Learning Approaches, since their primary focus lies in optimizing behavior through iterative learning—defining reward signals, training policies, fine-tuning, and incorporating preferences. By contrast, we reserve Agent-based Orchestration for frameworks emphasizing real-time decision-making, distinguishing between approaches centered on how models learn versus how they act in real time.

5.1.3. Contrastive Learning

Contrastive learning (Figure 8) is a training paradigm that sharpens a model's decision boundaries when it comes to differentiating between positive examples (factually sound statements) from negative examples (incorrect information). A major benefit of contrastive learning is its ability to use unlabeled or weakly labeled data and to efficiently learn representations from unstructured data as shown in techniques such as SimCLR [140] and Momentum Contrast (MoCo) [141]. Although initially targeting computer vision, these methods were later adapted to help mitigate hallucinations by generating synthetic "negative" examples, thus demonstrating that contrastive learning can significantly reduce the need for large and curated datasets [140,141].

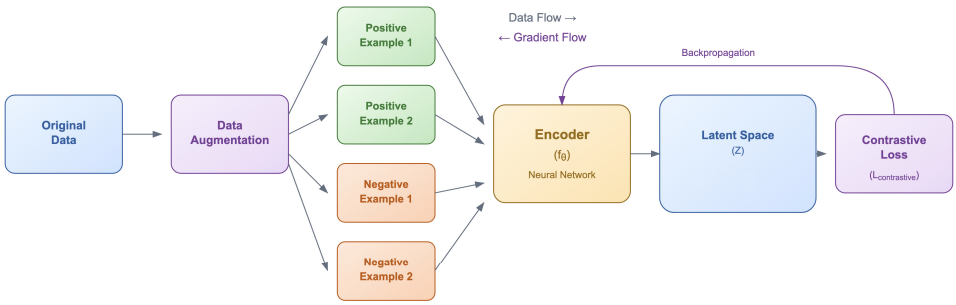

Figure 8. Illustration of the contrastive learning paradigm used to distinguish hallucinated from factual content through hard negative sampling and representation alignment.

Contrastive methods can improve content quality, such as in [142] where the loss is calculated over ranked candidate summaries and the model is penalized if a low-quality summary is selected over a better one. This loss is recombined with the standard MLE loss, thus resulting in the optimization of a combined, contrastive reward loss for factuality-ranked summaries. In contrastive learning, carefully constructed negative samples that closely resemble plausible but incorrect content (known as "hard negatives") can be particularly beneficial, as they challenge the model to learn subtle distinctions [143]. Although [143] does not directly target hallucinations, its proposed debiased contrastive loss has influenced research in hallucination mitigation that selects hard negatives for training models with stronger factual grounding such as [16,101].

Several studies have examined contrastive learning as a strategy for mitigating hallucinations in LLMs. MixCL reduces hallucinations in dialogue tasks by training models to distinguish between appropriate and inappropriate responses through negative sampling and mixed contrastive learning, though it faces challenges such as catastrophic forgetting [16]. This issue has been addressed with methods like elastic weight consolidation and experience replay, which preserve general knowledge by freezing core parameters while fine-tuning only on hallucination-prone tasks [61,122]. Other approaches, such as SimCTG, introduce contrastive objectives and decoding methods to improve coherence and reduce degeneracy [144]. Iter-AHMCL employs two guidance models—one trained on low-hallucination data and another on hallucinated samples—to provide iterative

contrastive signals that adjust LLM representation layers [145]. Contrastive Preference Optimization (CPO) combines negative log-likelihood loss with contrastive loss to favor non-hallucinated translations [101], and TruthX explicitly edits LLM parameters in "truthful space" using contrastive gradients, shifting activations toward truthful outputs [121]. These methods highlight the promise of contrastive learning for hallucination mitigation, though its effectiveness varies, with some studies suggesting that contrasting hallucinated versus ground-truth outputs may actually reduce detection accuracy [85].

While contrastive learning is most often used as a training method to refine model representations, many related strategies apply its core principle of contrasting desirable and undesirable outputs. For instance, Decoding by Contrasting Retrieval (DeCoRe) penalizes hallucination-prone predictions during inference [146], while Delta compares masked and unmasked inputs to suppress hallucinated tokens [147]. LLM Factoscope uses contrastive analysis of token distributions across layers to separate factual from hallucinated outputs [148], and Self-Consistent Chain-of-Thought Distillation (SCOTT) filters rationales from teacher models, retaining those most consistent with correct answers [149]. Decoding by Contrasting Layers (DoLa) contrasts early and late-layer activations to suppress hallucinated distributions, leveraging the observation that factual knowledge often resides in deeper layers [150]. Adversarial Feature Hallucination Networks (AFHN) combine adversarial alignment with an anti-collapse regularizer to prevent mode-collapsed feature hallucinations [90]. In neural machine translation, contrastive pairs of hallucinated vs. non-hallucinated outputs have also been used as supervision for hallucination detection, albeit without a contrastive loss [151]. Although most of these methods, apart from [90,151], operate without training, they illustrate how the contrastive paradigm can enhance generation quality, and we address them further in Section 5.2.2 "Decoding Strategies," Section 5.5.1 "Internal State Probing," and Section 5.1.4 "Knowledge Distillation."

5.1.4. Knowledge Distillation

Knowledge distillation (Figure 9) involves training a smaller and more efficient student model to emulate the outputs and learned representations of a larger, more knowledgeable teacher model thereby allowing the student model to benefit from the factual grounding and calibrated reasoning of the teacher [58,122,152]. Knowledge distillation (KD) has emerged as a promising approach to enhance the factual accuracy of LLMs, particularly in tasks susceptible to hallucinations, such as summarization and question answering.

The efficacy of knowledge distillation in reducing hallucinations is shown in studies such as [58], where transferring knowledge from a high-capacity teacher to a smaller student model yields significant improvements in exact match accuracy and reductions in hallucination rates. A smoothed knowledge distillation method is introduced in [152], where soft labels mitigate hallucinations and improve factual consistency across summarization and QA. Soft labels, viewed as probability distributions, train the student to be less overconfident and more grounded. Building on these findings, Ref. [122] compares knowledge distillation with self-training in retrieval-augmented QA, showing both to be beneficial but with knowledge distillation offering slightly stronger factual accuracy. Further extensions include Chain-of-Thought (CoT) Distillation and SCOTT, which leverage CoT reasoning and self-evaluation of larger models to train smaller models for multi-step reasoning without generating lengthy intermediate steps [149,153]. Similarly, Ref. [154] employs LLM-generated explanations to fine-tune a Small Language Model (SLM), followed by reinforcement learning to reward correct reasoning paths.

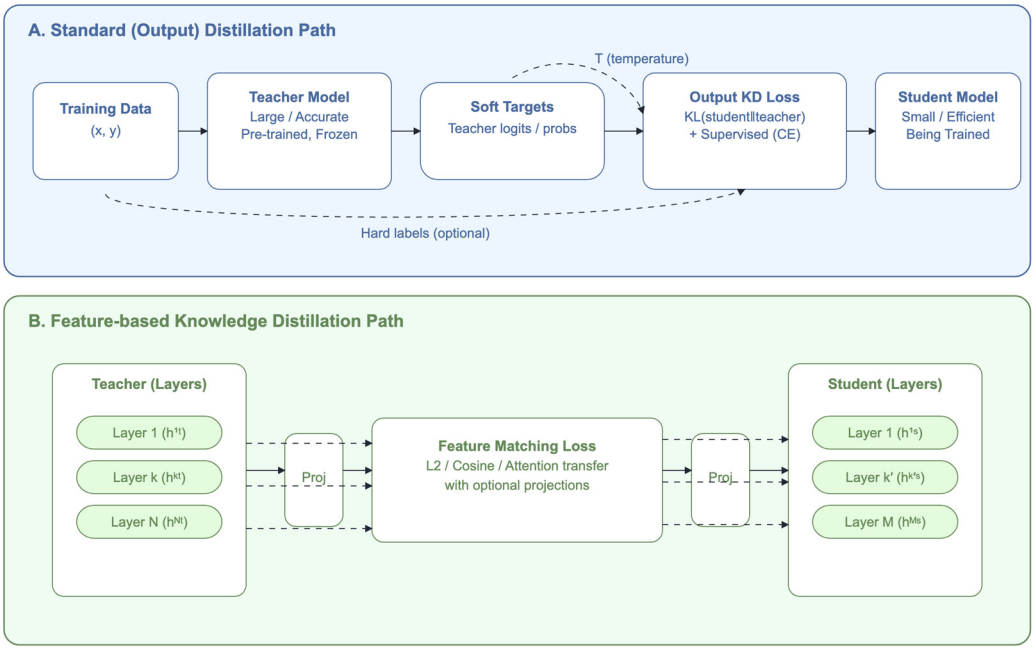

Figure 9. Knowledge distillation framework where a student model learns to emulate a teacher model's outputs to reduce hallucinations and improve factual consistency.

5.1.5. Instruction Tuning

Instruction tuning (Figure 10) is a specialized form of supervised learning that fine-tunes language models on datasets of input–output pairs framed as natural language instructions [128]. While it uses prompts for formatting, its focus is on training models to behave differently rather than relying on prompting at inference, distinguishing it from in-context methods discussed in "In-context Prompting." The goal is to optimize models to follow instructions more faithfully, improving factuality and coherence to carefully designed prompts [155,156], or explicitly encouraging factual grounding while discouraging speculation [15,99,157]. Instruction tuning also benefits directly from scaling [63,156], enhancing task generalization and mitigating hallucinations by aligning outputs with factual instructions [99,127,158]. Furthermore, it enables uncertainty-aware responses [155,159] through mechanisms such as:

- Factual alignment can be achieved through domain-specific fine-tuning, as in [99], where LLMs are trained on datasets of legal instructions and responses. Another approach, Curriculum-based Contrastive Learning Cross-lingual Chain-of-Thought (CCL-XCoT), integrates curriculum-based cross-lingual contrastive learning with instruction fine-tuning to transfer factual knowledge from high-resource to low-resource languages. Its Cross-lingual Chain-of-Thought (XCoT) strategy further enhances this process by guiding the model to reason before generating in the target language, effectively reducing hallucinations [160].
- Consistency alignment, which is achieved in [158] during a two-stage supervised fine-tuning process: The first step uses instruction–response pairs while in the second step, pairs of semantically similar instructions are used to enforce aligned responses across instructions.
- Uncertainty awareness, where techniques such as R-Tuning are employed to incorporate uncertainty-aware data partitioning, guiding the model to abstain from answering questions outside its parametric knowledge [15,155].
- Data-centric grounding, where Self-Instruct introduces a scalable, semi-automated method for generating diverse data without human annotation [159]. It begins with a small set of instructions and uses a pre-trained LLM to generate new tasks and

corresponding input-output examples, which are then filtered and used to fine-tune the model, thus generating more aligned and grounded outputs.

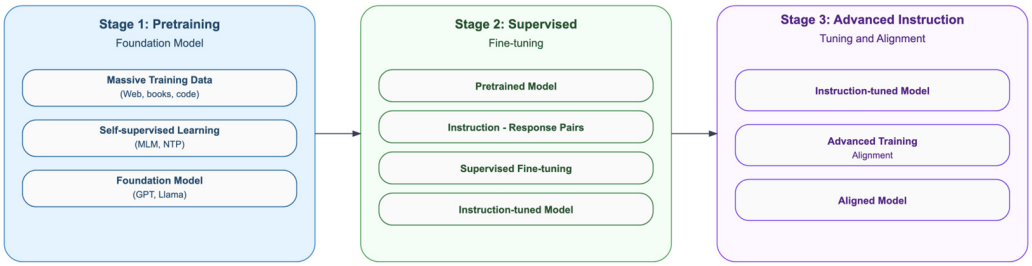

Figure 10. Instruction tuning paradigm showing how models are trained on natural language instructions to improve factual alignment and task generalization.

Research demonstrates that instruction tuning can be effectively scaled by curating a high-quality collection of over 60 publicly available datasets spanning diverse NLP tasks [156]. Using the Flan-PaLM model, substantial gains were reported in zero-shot and few-shot generalization, with results showing that greater instruction diversity improves factual grounding and reduces hallucinations on unseen tasks by exposing models to broader factual contexts [127,156]. Additionally, incorporating Chain-of-Thought (CoT) reasoning during training further strengthens reasoning capabilities [156].

Despite its benefits, instruction tuning often exacerbates the issue of hallucinations in LLMs since fine-tuning on unseen examples can increase hallucination rates without safeguards [89] and negatively impact the calibration of a model's predictive uncertainty leading to overconfident but hallucinated outputs [79], a finding which is also corroborated by independent observations from OpenAI in the GPT-4 "Technical Report" [131]. To mitigate these risks, techniques explicitly guide a model to say "don't know" or generate refusal responses during the instruction tuning process in order to encourage the model to express uncertainty when appropriate [15,155], or make use of more robust decoding strategies and uncertainty estimation methods [79].

5.2. Architectural Modifications

Architectural modifications refer to structural changes and enhancements in model design or inference processes aimed at improving performance, efficiency, and generation quality. These include advances in attention mechanisms, refined decoding strategies, integration of external information through Retrieval-Augmented Generation (RAG), and the use of alternative knowledge representation approaches or specialized mechanisms to improve controllability and factual accuracy. Since some methods bridge architectural and post-generation paradigms, we categorize them based on their primary operational stage: approaches that modify model internals or inference with structured knowledge are treated as "Architectural Modifications," while those that operate mainly on generated outputs for verification or attribution are discussed in "Post-Generation Quality Control".

5.2.1. Attention Mechanisms

The attention mechanism, first introduced in a groundbreaking paper on neural machine translation [161] and later revolutionized by the Transformer architecture [162], transformed natural language processing by allowing models to weigh different parts of the input context dynamically, thereby focusing on the most relevant information at each step of generation. This granular control can directly reduce hallucinations by helping the model pinpoint critical pieces of factual data while research on multi-head attention has shown that heads specialized in factual recall can guide the model to learn multiple relationships

in parallel [163]. Research links hallucinations to shifts in attention weights, where input perturbations can steer models toward specific outputs [164]. Attention mechanisms also help explain instruction drift, as attention decays over extended interactions [165]. Split-softmax addresses this by reweighting attention toward the system prompt, mitigating drift-related hallucinations [165]. Inference-Time Intervention (ITI) identifies attention heads containing desired knowledge and modifies their activations, with nonlinear probes extracting additional information to guide more factual outputs [166]. Other approaches adapt attention using Graph Neural Networks for better contextual focus [167], Sliding Generation with overlapping prompt segments to reduce boundary information loss [101], and novel head architectures [168].

Another line of research focuses on directly manipulating tokens. Adaptive Token Fusion (ATF) merges tokens to alter the weight distribution in attention, conditioning computation over a more efficiently structured token space [169]. Truth Forest applies multi-dimensional orthogonal probes to detect "truth representations" in hidden states, shifting them along directions of maximum factuality to bias the attention mechanism, while Random Peek extends this across wider sequence positions to reinforce truthfulness and reduce hallucinations [120]. Other methods indirectly manipulate attention by conditioning the encoder or decoder during computation. For example, Ref. [170] analyzes attention patterns in relation to factual correctness, modifying them through a decoding framework, while [146] perturbs hallucinations by masking attention heads that specialize in content retrieval, conditioning the contrastive decoding process. In [171], an NLU model parses meaning-representation (MR) pairs from utterances and reconstructs correct attribute values in reference texts, using a self-attentive encoder to produce slot-value/utterance vectors and an attentive scorer to compute semantic similarity, followed by self-training to recover semantically aligned texts. Language-specific modules have also been introduced to coordinate encoder and decoder activation, restricting them to intended input and target languages [118]. Finally, attention maps serve as diagnostic signals for factuality in approaches like AggTruth [172], which aggregates attention across layers to compute contextualized patterns and detect hallucinations in RAG settings.

5.2.2. Decoding Strategies

In this section, we present decoding strategies as intervention methods applied during the generation process of LLMs, determining how models translate their learned representations into human-readable text by balancing fluency, coherence, and factual accuracy. Hallucinations often arise because models fail to adequately attend to input context or rely on outdated or conflicting prior knowledge [173]. While standard decoding methods such as greedy decoding, beam search, and nucleus sampling [75] offer trade-offs between diversity, efficiency, and precision [5], models can still commit to incorrect tokens early in decoding and subsequently justify them, reflecting fundamental issues in decoding dynamics [21,65]. To address this, decoding strategies are frequently developed alongside complementary approaches, including attention mechanisms [118,146,170], probability-based methods [29,112,174,175], or RAG-based decoding [154,176], all aiming to enhance output quality and interpretability. Empirical evidence shows that advanced decoding strategies can substantially reduce hallucinations: for example, RAG raises FactScore from 17.5% to 42.1%, while self-correction methods have improved truthfulness scores on benchmarks like TruthfulQA [5]. Building on these insights, we categorize advanced decoding strategies as follows:

Methods in this family adjust token selection to favor context-aligned, higher-confidence outputs. Context-aware decoding amplifies the gap between probabilities with vs. without context, down-weighting prior knowledge when stronger contextual

evidence is present [29,173]. Entropy-based schemes penalize hallucination-prone tokens using cross-layer entropy or confidence signals [174], while CPMI rescales next-token scores toward tokens better aligned with the source [175]. Logit-level interventions refine decoding by interpreting/manipulating probabilities during generation [112]. Confidence-aware search variants use epistemic uncertainty to steer beams toward more faithful continuations, with higher predictive uncertainty correlating with greater hallucination risk [76,177]. SEAL trains models to emit a special [REJ] token when outputs conflict with parametric knowledge and then leverages the [REJ] probability at inference to penalize uncertain trajectories [178]. Finally, factual-nucleus sampling adapts sampling randomness by sentence position, substantially reducing factual errors [116].

- Probabilistic Refinement and Confidence-Based Adjustments: Methods in this family adjust token selection to favor context-aligned, higher-confidence outputs. Context-aware decoding amplifies the gap between probabilities with vs. without context, down-weighting prior knowledge when stronger contextual evidence is present [29,173]. Entropy-based schemes penalize hallucination-prone tokens using cross-layer entropy or confidence signals [174], while CPMI rescales next-token scores toward tokens better aligned with the source [175]. Logit-level interventions refine decoding by interpreting/manipulating probabilities during generation [112]. Confidence-aware search variants—Confident Decoding and uncertainty-prioritized beam search—use epistemic uncertainty to steer beams toward more faithful continuations, with higher predictive uncertainty correlating with greater hallucination risk [76,177]. SEAL trains models to emit a special [REJ] token when outputs conflict with parametric knowledge and then leverages the [REJ] probability at inference to penalize uncertain trajectories [178]. Finally, factual-nucleus sampling adapts sampling randomness by sentence position, substantially reducing factual errors [116].
- Contrastive-inspired Decoding Strategies: A range of decoding methods build on contrastive principles to counter hallucinations. DeCoRe induces hallucinations by masking retrieval heads and contrasting outputs of the base LLM with its hallucination-prone variant [146], while Delta reduces hallucinations by masking random input spans and comparing distributions from original and masked prompts [147]. Contrastive Decoding replaces nucleus or top-k search by optimizing the log-likelihood gap between an LLM and a smaller model, introducing a plausibility constraint that filters low-probability tokens [17]. SH^2 (Self-Highlighted Hesitation) manipulates token-level decisions by appending low-confidence tokens to the context, causing the decoder to hesitate before committing [179]. Spectral Editing of Activations (SEA) projects token representations onto directions of maximal information, amplifying factual signals and suppressing hallucinatory ones [180]. Induce-then-Contrast (ICD) fine-tunes a "factually weak LLM" on non-factual samples and uses its induced hallucinations as penalties to discourage untruthful predictions [181]. Active Layer Contrastive Decoding (ActLCD) applies reinforcement learning to decide when to contrast layers, treating decoding as a Markov decision process [182]. Finally, Self-Contrastive Decoding (SCD) down-weights overrepresented training tokens during generation, reducing knowledge overshadowing [54].
- Verification and Critic-Guided Mechanisms: Several strategies enhance decoding by incorporating verification signals or critic models. Critic-driven Decoding combines an LLM's probabilistic outputs with a text critic classifier that evaluates generated text and steers decoding away from hallucinations [115]. Self-consistency samples multiple reasoning paths, selecting the most consistent answer; this not only improves reliability but also provides an uncertainty estimate for detecting hallucinations [21]. TWEAK treats generated sequences and their continuations as hypotheses, which are

reranked by an NLI or Hypothesis Verification Model (HVM) [119]. Similarly, mFACT integrates a faithfulness metric into decoding, pruning candidate summaries that fall below a factuality threshold [100]. RHO (Reducing Hallucination in Open-domain Dialogues) generates candidate responses via beam search and re-ranks them for factual consistency by analyzing knowledge graph trajectories from external sources [183].

- Internal Representation Intervention and Layer Analysis: Understanding how LLMs encode replies in their early internal states is key to developing decoding strategies that mitigate hallucinations [30]. Hallucination-prone outputs often display diffuse activation patterns rather than concentrated on relevant references. In-context sharpness metrics address this by enforcing sharper token activations, ensuring predictions emerge from high-confidence knowledge areas [184]. Inference-Time Intervention (ITI) shifts activations until the response is complete [185], while DoLa contrasts logits from early and later layers, emphasizing factual knowledge embedded in deeper layers over less reliable lower-layer signals [150]. Activation Decoding similarly constrains token probabilities using entropy-derived activations without retraining [184]. LayerSkip introduces self-speculative decoding, training models with layer dropout and early-exit loss so that earlier layer predictions are verified by later ones, thereby improving efficiency [186].
- RAG-based Decoding: RAG-based decoding strategies integrate external knowledge to enhance factual consistency and mitigate hallucinations [154,176]. For instance, REPLUG prepends a different retrieved document for every forward pass of the LLM and averages the probabilities from these individual passes, thus allowing the model to produce more accurate outputs by synthesizing information from multiple relevant contexts simultaneously [176]. Similarly, Retrieval in Decoder (RID) dynamically adjusts the decoding process based on the outcomes of the retrieval, allowing the model to adapt its generation based on the confidence and relevance of the retrieved information [154].

5.2.3. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) (Figure 11) modifies the architecture or inference pipeline by including a retrieval component. Given an initial prompt, this component fetches the most relevant documents from external sources and injects them into the prompt to enrich the prompt's context [17]. This extends an LLM's access beyond its training data, grounding outputs in updated references and reducing hallucinations across diverse tasks [17,40,187]. Most implementations follow a pre-LLM refine-then-generate paradigm, where retrieved documents enrich the prompt before generation, while post-LLM generate-then-refine approaches first produce an intermediate response to guide retrieval, enabling refinement with highly relevant information [28,61,188–190]. Retrieval typically uses dual encoders, semantic indexing, or vector similarity search [191].

Many studies analyze how retrieval choices impact RAG performance. In [192], dense and sparse retrieval are compared, and a hybrid retriever is proposed that combines both with advanced query expansion; results are fused via weighted Reciprocal Rank Fusion. HaluEval-Wild constructs reference documents from the web using both dense and sparse retrievers to assess whether model outputs contain hallucinations [107]. HalluciNot repurposes RAG for detection: it retrieves external documents, trains a classifier on them, and then verifies whether each generated span is supported by factual evidence [111]. While RAG has proven effective in mitigating hallucinations through external source grounding, research also shows that, paradoxically, hallucination rates may increase under certain conditions. To better understand these challenges, we outline in Section 7.2 a set of practical considerations for users when applying RAG-based methods. These include:

- Dependency on Retrieval Quality: RAG-based methods rely heavily on the quality of retrieved documents and so inaccurate, outdated, noisy, or irrelevant retrievals may mislead the language model or introduce additional hallucinations [51,55,57,62,73,107,193–195]
- Complex Reasoning: Retrieval systems may often strive to engage in complex or multi-hop reasoning thus harming performance of natural language inference (NLI) models [51,57,62].
- Latency and Scalability: Lengthy prompts and iterative prompting improve factuality but at the cost of increased inference time and resource consumption, reducing scalability in real-world settings [51,193,194].
- Over-reliance on External Tools: Some systems require dependency on entity linkers, search engines, or fact-checkers, which can introduce errors or inconsistencies [51,62,109].
- Faithfulness and effective integration: Methods that prioritize factuality sometimes harm the fluency or coherence of generated text, leading to less natural outputs [51] while integrating it effectively into the generation process is another important factor [57].

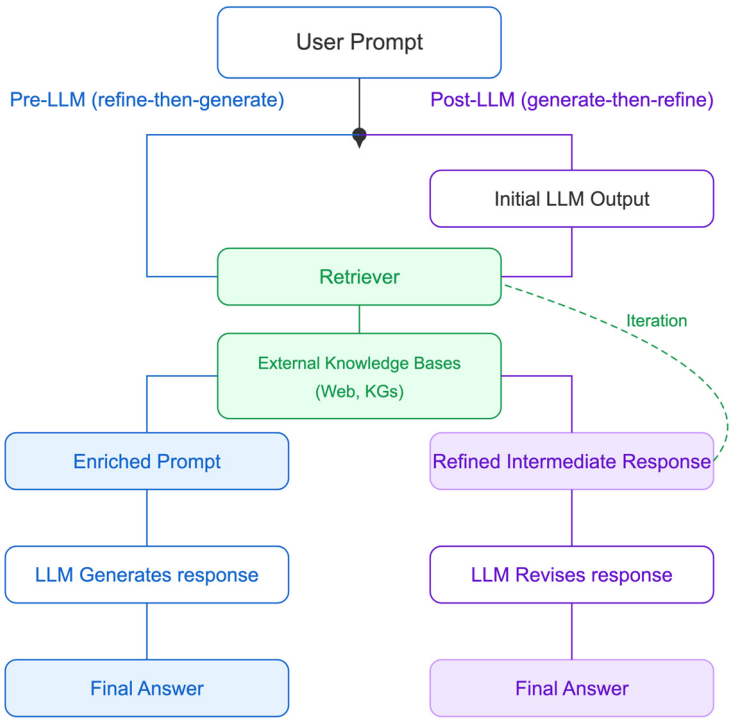

Figure 11. Simplified overview of Pre-LLM and Post-LLM retrieval-augmented generation (RAG) pipelines.

Several extensions strengthen RAG by improving how evidence is gathered, structured, and applied. LLM-AUGMENTER retrieves external evidence, consolidates it via entity linking and evidence chaining, and feeds the structured evidence into the LLM; it iteratively flags factual inconsistencies and revises prompts until outputs align with retrieved knowledge [193]. CRAG leverages Wikipedia and Natural Questions with BM25 retrieval, using both the original query and segments of the LLM's draft to fetch supporting passages [194]. SELF-RAG introduces a self-reflective loop in which a critic emits reflection tokens and the generator conditions on them, enabling adaptive retrieval, self-evaluation, and critique-guided decoding [196]. FreshLLMs augment with web search to detect and correct factual errors using retrieved evidence, addressing outdated parametric knowledge [62]. In [154], Mistral is modified to integrate a retrieval module that dynamically fetches top-k Wikipedia documents and generates fact-grounded responses. Finally,

AutoRAG-LoRA combines prompt rewriting, hybrid retrieval, and hallucination detection to score confidence and trigger selective LoRA adapter activation [197].

The effectiveness of RAG depends heavily on the quality of retrieved content, yet this assumption of noise-free evidence does not always hold. To address noisy retrieval, Ref. [73] proposes a two-stage approach: filtering irrelevant documents with a trained classifier, followed by contextual fine-tuning that conditions the model to disregard misleading inputs. Beyond noise, the timing of retrieval has emerged as equally critical, since unnecessary retrieval can introduce irrelevant information and increase both inference time and computational cost [55]. DRAD tackles this with real-time hallucination detection (RHD) and a self-correction mechanism using external knowledge to dynamically adjust retrieval. ProbTree integrates retrieved and parametric knowledge via a hierarchical reasoning structure with uncertainty estimation [198]. Recent work has also advanced document preprocessing: a semantic chunker improves indexing by ensuring that retrieved segments are contextually meaningful and better aligned with the generation process [6].

In addition to addressing the challenges of noisy retrieval, recent research has explored a variety of new frameworks to optimize RAG-based systems. Read-Before-Generate (RBG) pairs an LLM with a reader that assigns sentence-level evidence probabilities, fuses multiple retrieved documents at inference, and uses a specialized pre-training task to promote reliance on external evidence [199]. REPLUG retrieves with a lightweight external module, prepends documents to the prompt, and reuses the same LLM to encode them [176]. Neural-retrieval-in-the-loop targets complex multi-turn dialogue, with RAG-Turn enabling synthesis across multiple documents so retrieval stays relevant to each turn, mitigating topic drift and factual errors [74]. Retrieval in Decoder (RID) integrates retrieval directly into decoding so that generation and retrieval proceed jointly [154]. Finally, SEBRAG performs substring-level extraction conditioned on document ID and span boundaries, enabling precise answers and explicit abstention when there is no answer [200].

RAG-based methods have been used in dataset creation and code generation. For instance, ANAH provides an analytically annotated hallucination dataset, where each answer sentence is paired with a retrieved reference fragment used to judge hallucination type and correct errors—highlighting the role of retrieval in annotation [57]. ChitChatQA [109] assesses LLMs on complex, diverse questions, built through a framework that invokes tools and integrates retrieved information from reliable external sources into reasoning. In code generation, RAG-based methods retrieve relevant code snippets from repositories based on task similarity, incorporating them into prompts and achieving consistent performance gains across evaluated models [66].

An increasing number of studies apply RAG to retrieve specific information from Knowledge Graphs, so as to enrich the initial prompt's context-specificity. For instance, the Recaller module [201] retrieves entities via a two-stage highlighting process: a coarse filter flags potential hallucinations, while a fine-grained step masks and regenerates them. Graph-RAG [195] combines HybridRAG, which injects graph entities or relations into context based on semantic proximity or relation paths, with FactRAG, which evaluates factual consistency and filters hallucinated responses. Other methods use graph-based detection by extracting structured triplets from LLM outputs and performing explicit relational reasoning to compare them with reference knowledge graphs, thereby identifying unsupported or fabricated facts [202]. While most RAG methods retrieve before generation, several approaches use retrieval after generation to validate and correct outputs. A post hoc RAG pipeline in [61] extracts factual claims from model outputs and verifies them against knowledge graphs; RARR identifies unsupported claims, finds stronger evidence, and revises text accordingly [190]. Neural Path Hunter reduces dialogue hallucinations by

traversing a knowledge graph to build a KG-entity memory and using embedding-based retrieval during refinement, even without a standard RAG architecture [188].

5.2.4. Knowledge Representation Approaches

Knowledge representation approaches (Figure 12) incorporate structured and semi-structured knowledge, such as knowledge graphs [88,188,201,202], ontologies [203], or entity triplets into a language model's architecture to enhance reasoning and factual consistency [204]. These representations emphasize explicit connections, such as hierarchical classifications and causal links, thereby enabling the verification of statements against known relationships. To ensure that the integration of such knowledge representations provides satisfactory results, it is necessary to maintain high-quality and up-to-date content, which is readily available and free from ambiguities [73,201,204].

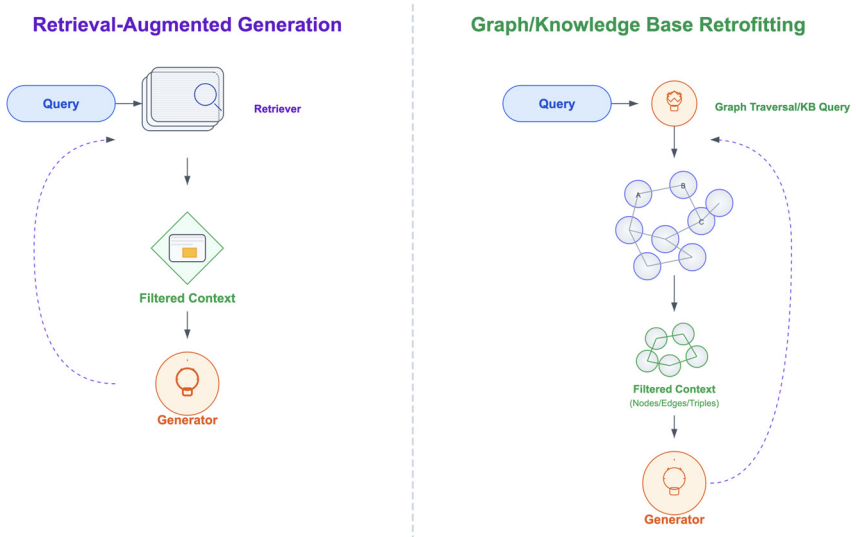

Figure 12. Graph/Knowledge Base injection vs. RAG diagram, effectively showing how these approaches process queries differently.

Integration methods span prompting during inference [205], retrofitting external knowledge [61,195,201,206], modifying the attention mechanism [167], and directly injecting graph information into latent space [202,204]. The Recaller module [201] retrieves entities and their one-hop neighbors from a knowledge graph and guides generation in two stages: a coarse filter identifies potentially hallucinated spans, and a fine-grained step selectively masks and regenerates them, enabling more precise grounding in retrieved knowledge. This intervention improves factual accuracy while maintaining fluency, though retrievers may lack sufficient relational depth for complex queries—a limitation also noted in [73]. FLEEK [207] follows a related path by extracting factual claims, converting them into triplets, and generating questions for each triplet, which are answered via a KG-based QA system to verify the extracted facts.

Knowledge graphs (KGs) serve as external sources to enrich context before generation [88,189,195,203,204]. Graph-integration techniques include entity–relation or subgraph extraction and multi-hop reasoning via traversal [195,208], often linearized into hierarchical prompts or encoded to fit LLM inputs [208]. In [195], modular retrieval expands or re-ranks passages using entity linking, subgraph construction, or semantic path expansion, leveraging graph-based signals (e.g., proximity, relation paths) to improve factual grounding. Similarly, Ref. [203] evaluates DBpedia as an external source, concluding that while LLMs encode factual knowledge in parametric form, this implicit knowledge is limited, and external KGs remain essential. Knowledge Graph Retrofitting [61] validates

only critical information extracted from intermediate responses, then retrofits KG evidence to refine reasoning. LinkQ [209], an open-source NLI interface, directs LLMs to query Wikidata, outperforming GPT-4 in several scenarios though struggling with complex queries. ALIGNed-LLM [210] integrates entity embeddings from a KGE model via a projection layer, concatenated with token embeddings, reducing ambiguity and enabling structurally grounded outputs. G-retriever [211] addresses context length constraints using soft prompting and Prize-Collecting Steiner Tree optimization to select subgraphs balancing node relevance and connection cost, thus improving retrieval quality and efficiency.

A closely related line of research focuses on embedding graph-related information directly into the latent space of a model. Approaches leverage Graph Attention Networks (GAT) and message passing to explore semantic relations among hallucinations, assuming they are structured and follow homophily—entities with similar traits tend to connect [202]. Graph Neural Networks (GNNs) integrate with the attention mechanism to form hybrid architectures that aggregate information from neighboring nodes, enhancing contextual understanding and reducing hallucinations [167]. Rho improves open-domain dialogue by grounding parametric knowledge in retrieved embeddings (local knowledge) and augmenting attention with multi-hop reasoning over linked entities and relation predicates (global knowledge), thereby generating less hallucinated content [183].

5.2.5. Specialized Architectural Mechanisms for Enhanced Generation

This category encompasses advanced frameworks that we have found difficult to classify in the previous sections of "Architectural Modifications". A number of techniques—including diffusion-inspired mechanisms, Mixture-of-Experts frameworks [40], Ref. [212], dual-model architectures [213], LoRA adapter integration and specialized classifiers [40,111], token pruning and fusion [169,214], and Representation Engineering [112], have demonstrated significant potential in addressing hallucinations.

MoE architectures, introduced in [215] and scaled for LLMs in [216], partition models into expert modules, with a gating network selecting a few experts per input for sparse, efficient activation. By localizing knowledge and reducing interference across domains, MoEs can help mitigate hallucinations, though they typically increase parameter count. Extending this idea, Lamini-1 [40] implements a Mixture of Memory Experts (MoME) with millions of experts acting as a structured factual memory, enabling explicit storage/retrieval of facts during inference via sparse expert routing, and combining cross-attention routing, LoRA-style adapters, and accelerated kernels. Similarly, the MoE in [212] orchestrates specialized experts through input preprocessing, dynamic expert selection, decision-routing to control information flow, and majority voting—filtering erroneous responses while containing computational overhead.

In addition to MoE architectures, dual-model architectures have also been used to effectively combat hallucinations as demonstrated in [213] where the primary model generates the initial text based on the input prompt, the secondary model monitors and analyzes the output from the primary model in real-time, checking for logical and factual inaccuracies while a feedback loop mechanism further enhances the self-monitoring process by allowing for iterative improvements based on detected errors. This interplay allows the system to cross-reference its outputs, thus improving the reliability of the generated text since any inconsistencies are immediately addressed [213]. A multi-task architecture for hallucination detection, HDM-2 [111], uses a pretrained LLM backbone with specialized detectors—e.g., a classification head for common-knowledge errors—plus LoRA-adapted components and shallow context-based classifiers, allowing tailored factual verification. The RBG framework [199] jointly models answer generation and machine reading via a reader that assigns evidence probability scores and injects them into the generator's

final distribution; a factual-grounding pretraining objective further encourages reliance on retrieved documents to boost accuracy. Rank-adaptive LoRA fine-tuning (RaLFiT) [217] focuses parameter updates on truthfulness-relevant modules: it first probes each transformer module for truthfulness correlation, then assigns higher LoRA ranks to the most correlated ones, and finally fine-tunes with Direct Preference Optimization (DPO) on paired truthful vs. untruthful responses to strengthen factual alignment.

Beyond architectural and dual-model designs, pruning and token fusion target hallucinations by restructuring computation. Pruning removes redundant weights, pushing the model to rely more on the source document and reducing unsupported content; Ref. [214] evaluates layer-wise magnitude, SparseGPT, and Wanda, finding pruned models show greater source reliance, higher lexical overlap with the source, and fewer hallucinations—though sometimes at the cost of contextual nuance. Adaptive Token Fusion (ATF) inserts an intermediate layer after tokenization and embeddings to dynamically alter token processing at inference [169]: it fuses redundant tokens by contextual similarity and prioritizes only salient tokens, yielding "semantic condensation" that lowers hallucination risk while also speeding inference and reducing compute.

Spectral Editing of Activations (SEA) is a training-free method that edits internal activations via spectral decomposition, projecting inputs toward directions that maximize covariance with positive demonstrations (truthful content) and minimize covariance with negative demonstrations (hallucinated content). By decorrelating activations from hallucinated examples, SEA steers generation toward more factual, reliable outputs [180]. Representation Engineering (RepE) treats representations as the core unit, improving interpretability and control. By tracking how concept representations evolve across layers, RepE enables targeted control of phenomena such as honesty vs. deception, reducing hallucinations. It employs Linear Artificial Tomography (LAT) to align representations with desired behaviors and deliver focused quality improvements [112]. TruthX is an inference-time intervention that uses an auto-encoder to map internal states into semantic and truthful latent spaces, and applies contrastive learning to discover "truthful editing directions." During generation, it shifts internal representations along these directions, increasing truthfulness and reducing hallucinations without retraining [121].

Finally, inspired by the denoising diffusion paradigm, recent research has explored architectural modifications collectively framed as latent diffusion, including improved attention mechanisms, hierarchical processing layers, and enhanced control over the propagation of latent representations within the model. For example, in [218], a modified Mistral model that incorporates such latent diffusion components shows reduced hallucination rates, improved contextual coherence, and greater predictive accuracy.

5.3. Input/Prompt Optimization

Input/Prompt Optimization refers to strategies for carefully crafting and refining the text provided to AI models to steer their behavior and output, often specifically to mitigate hallucinations. This includes comprehensive Prompt Engineering techniques like Structured or Iterative Reasoning Prompting (e.g., Chain of Thought) which can expose potential flaws in logic leading to hallucinations, and In-context Prompting using carefully synthesized prompts in order to guide the model away from hallucinations. It also involves Context Optimization where techniques such as RAG and meta-prompting optimize prompts and System Prompt Design to specifically synthesize persistent prompts that guide the generation process and alignment.

5.3.1. Prompt Engineering

Prompt engineering involves the strategic design of input prompts to constrain the generative space, leveraging LLMs' sensitivity to contextual cues [46,70,71]. This section covers general-purpose prompting for hallucination mitigation; specialized forms—in-context prompting, system prompt design, and meta-prompting—are treated elsewhere. The core premise is that even subtle prompt edits can narrow the search space and shift the output distribution [46], yielding sizable performance gains [70] and lowering hallucination rates [71]. Prompts may be explicit instructions or contextual scaffolds, ranging from simple rephrasings to constraints, or requests to verify information or cite sources [219]. Their influence is especially strong in zero-/few-shot settings [20,220], and prompt effects remain an active research topic in NLP [221–223]. Even work on architectural changes (e.g., pruning) controls for prompt sensitivity by testing multiple templates to isolate causal effects on hallucinations [214]. Despite its impact, prompt engineering is still ad hoc, relying on experimentation and domain expertise [72], motivating systematic approaches such as algorithmic prompt search/generation [102] and pattern catalogs of reusable strategies [221].

In guiding content generation and self-correction, prompt engineering can also be used to instruct LLMs such as ChatGPT to create, filter, and vary hallucinated questions based on specific constraints, explicitly guiding the model's generation behavior, style and factual correctness across different sampling strategies [71]. Yet a key limitation persists: models often default to familiar phrasings due to entrenched patterns, curbing the gains from careful prompt design—a finding echoed in [1,221,224]. Self-reflection prompts probe internal beliefs by asking models first to generate statements and then to assess their truthfulness, which can expose contradictions indicative of hallucinations [225]. However, such prompt-based introspection is unreliable: models frequently evaluate their own outputs inconsistently or overestimate correctness [225], and prior work likewise shows that self-evaluation does not guarantee faithful assessment [119,153,213].

Prompt engineering has been used to improve factual extraction and recall. In clinical IE, carefully crafted prompts with exhaustive researcher-generated search terms achieved near-perfect agreement with a string-search baseline, though they required expertise and were time-consuming [47,72]. For knowledge-graph entities, task-specific prompts were formatted to mirror triples and paired with few-shot examples to prime accurate tail prediction, improving consistency in KG completion tasks [203]. However, performance dropped on long-tail entities, indicating that prompt design alone cannot overcome inherent knowledge gaps or biases toward over-represented entities [203,204]. These limitations are especially consequential in high-stakes domains such as legal research, where LLMs may produce hallucinated cases, statutes, or citations with high confidence [8].

In addition to improving factual extraction and recall, prompt engineering is also used to deliberately induce or surface hallucinations in GPT [46] or to elicit lies via unrelated or tangential questions that expose internal inconsistency patterns [117]. Varying prompt phrasing and posing specially designed probes to test the truthfulness of earlier outputs is effective for assessing how responses shift under challenge; however, lie detection is highly sensitive to prompt wording (see word sensitivity) and its success depends heavily on how questions are framed, limiting generalization across prompt variants and domains [70,72,117,226]. These phrasing idiosyncrasies also appear in negation-related tasks, where cautionary instructions and in-context exemplars can guide models toward more accurate outputs [227]. Yet, when an LLM is queried with false-premise prompts, available contextual knowledge may increase hallucinations, and self-refinement can paradoxically amplify them rather than mitigate them [67,227].

Beyond being used as a standalone intervention, prompt engineering also acts as a vital enabling component in a wide array of hallucination mitigation strategies which include (but are not limited to): supervised and semi-supervised learning pipelines, decoding strategies, RAG, reinforcement learning, and agent-based architectures. Specifically:

- In dataset creation and evaluation, prompt engineering has been used to generate and filter references used for inference and evaluation [107,220,224,228], systematically induce, detect, or elicit imitative falsehoods [42,53,93,117,139,181,202,223], and even create specific types of code hallucinations to test research methodologies [229,230].
- For confidence assessment and behavioral guidance, it has been used to elicit verbalized confidence, consistency, or uncertainty, and test or guide model behavior and alignment [4,22,30,71,77,95,97,103,120,179,194,203,225,227], reduce corpus-based social biases [80], extract and verify claims [231] as well as investigate failure cascades like hallucination snowballing [65].
- In knowledge integration scenarios it has been combined with retrieval modules or factual constraints [17,73,114,232], in agentic environments where prompts guide the generation of states, actions, and transitions [233], or the alignment process between queries and external knowledge bases [205], and even in the training process of a model where they are used to inject entity summaries and triplets [104]. Additionally, prompts have also been explored as explicit, language-based feedback signals in reinforcement learning settings, where natural language instructions are parsed and used to fine-tune policy decisions during training [69].

Some forms of prompt engineering go beyond shaping outputs or guiding behavior and instead aim to iteratively decompose prompts or elicit structured, step-by-step reasoning within the model. We will discuss such structured and iterative varieties of Prompt Engineering, such as Chain of Thought (CoT) [20] or Tree of Thoughts (ToT) [234], more extensively in the Section 5.3.2 "Structured or Iterative Reasoning Prompting". While prompt engineering is one of the most powerful methods for mitigating hallucinations, it also suffers from limitations and drawbacks such as:

- wording sensitivity where minor variations in wording can trigger different learned associations [70,72] or even amplify hallucinations [65],
- over-specificity where a model fails to generalize beyond the exact pattern that a prompt encodes [20,235], thus limiting mitigation generalizability,
- scalability issues which arise from the number of intermediate tasks or their complexity [235],
- context dilution which demonstrates that prompts often fail when irrelevant context is retrieved, especially in RAG scenarios [73],
- bias amplification as a negative side effect of alignment objectives and preferences [15,69,81],
- false confidence which occurs when prompts eliciting confidence may yield overconfident incorrect answers [77,83], thus undermining trust in self-correction mechanisms,
- lack of standardized prompting workflows which makes prompt engineering a significant trial and error task not only for end-users but also for NLP experts [72], hindering reliable mitigation, and
- jailbreaking, where adversarial prompt techniques are used to manipulate a model into generating content beyond its capabilities, thus frequently resulting in severe hallucinated content or alignment violations [219,236].

5.3.2. Structured or Iterative Reasoning Prompting

Research shows that prompt engineering can unlock LLMs' reasoning abilities, motivating a category we call Structured or Iterative Reasoning Prompting [20,234] (Figure 13).

These techniques explicitly scaffold the model's reasoning by prompting it to externalize and systematically explore intermediate steps before committing to a final answer. In contrast to static prompts, they aim to reduce hallucinations by decomposing tasks into verifiable steps and enabling multi-path exploration or self-correction [21]. Specifically:

- Structured reasoning prompts modify behavior within a single forward pass. The model follows the request to enumerate steps "in one shot"; typically, there is no separate controller deciding when to take steps or whether to call external tools.
- Iterative reasoning further improves generation by guiding decomposition into a series of steps, each of which builds on, refines, and supports previous steps before producing the final answer.

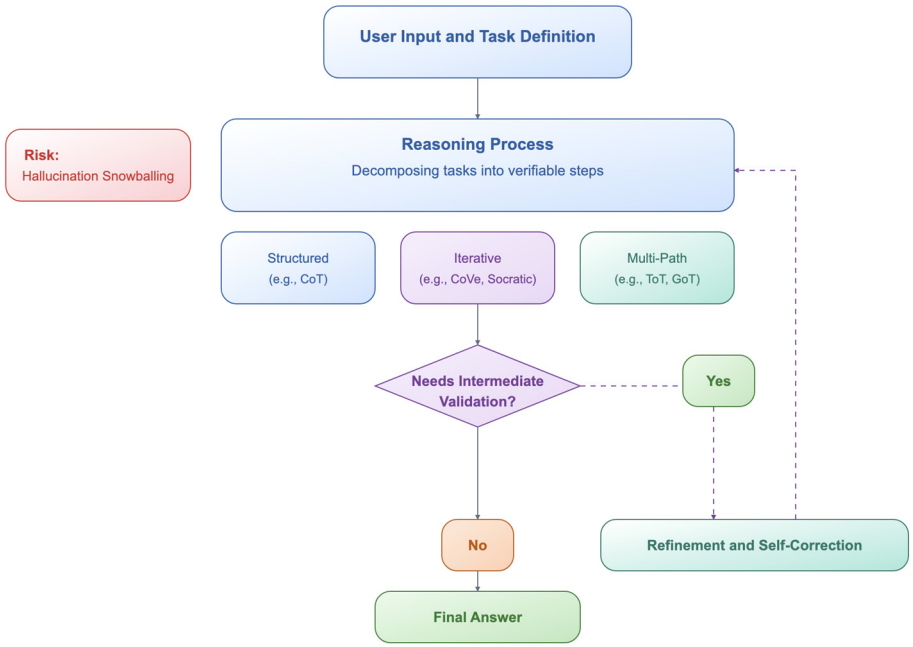

Figure 13. Flowchart illustrating structured, iterative, and multi-path reasoning prompting.

Techniques exemplifying these approaches and shown to mitigate hallucinations include Chain of Thought (CoT) prompting [20], Chain of Natural Language Inference (CoNLI) [220], and Chain of Verification (CoVe) [237]; Tree of Thoughts (ToT) further extends them by branching the reasoning process into multiple solution paths [234] (also supported in a self-debugging context [198]), while Graph of Thoughts (GoT) organizes intermediate steps into arbitrary graph structures [64]. Additionally, supplying an informal logical description of the target algorithm—combined with explanatory prompting and graph connectivity—significantly reduces hallucinations [238]. By exploring branches in parallel or sequentially, models can compare, refine, and discard suboptimal lines of thought to reach more reliable conclusions. However, CoT-based reasoning can still mislead, exhibiting logical inconsistencies that obscure why a prediction was made [70], and sometimes producing "absurd claims" or incorrect causal answers, especially in complex logical tasks [239]. Iterative methods also tend to benefit larger models more [63], with smaller models often requiring distillation to realize gains [149], and they introduce higher latency and computational cost, creating efficacy–efficiency trade-offs [234,237,240]. Despite these drawbacks, the sequential or parallel exploration of multiple intermediate steps can be especially useful for complex, multi-step tasks such as:

- to iteratively decompose compound statements, generate sub-statements from the original output, check for logical inconsistencies, or synthesize logical

proofs [20,67,109,198,220,241,242] or a controlled set of contrasting answers for expertise scoring [243].
- exploit the dialog capabilities of LLMs to detect logical inconsistencies by integrating deductive formal methods [82].
- use multiple CoT chains generated by an LLM to imbue smaller models with distilled knowledge and enhance comprehensive thinking [149,153]; systematically create datasets by injecting hallucinations used in training models [147] and mitigate them via reasoning [147,224].
- check and evaluate factual consistency between source and claims using iterative feedback [45,193], structured prompts [5,98,244], generate clarifying questions and correlate constraints between the query knowledge bases [205], or use retrieval and graph-based retrofitting approaches [61].
- verify or reason over intermediate steps so as to synthesize a final response [237,245], or use reasoning to introduce contextually relevant and up-to-date evidence [62] and rectify or refine this response across multiple steps [244,246,247].
- uncover why hallucinations occur by analyzing potential "risk factors" and attributing hallucination behavior to deficiencies in specific model capabilities like commonsense memorization, or memorization of disambiguation patterns [68], relational reasoning, and instruction following [248].

Despite improving reasoning transparency, Structured or Iterative Reasoning Prompting can unintentionally amplify hallucinations when early steps are flawed, leading to hallucination snowballing [65]. Comparative studies of direct prompts vs. zero-shot CoT show that prompt design directly influences both the trajectory and severity of hallucinated content. While many approaches rely on fixed templates or scaffolds [20,234,237], others adopt flexible, dialogic formats that simulate iterative reasoning. Socratic prompting repeatedly questions, challenges, or guides the model through multi-turn dialectic to reduce hallucinations via iterative self-correction and critique [249]. InterrogateLLM similarly reformulates the query repeatedly before evaluating the final answer for potential hallucinations [250]. In an adversarial dual-model setting, interrogation proceeds through multi-turn interactions where generated claims and targeted questions are iteratively juxtaposed to expose factual errors and inconsistencies [251].

We clarify that "step-by-step" reasoning can occur in both categories, but our taxonomy distinguishes them by the locus of control: when the reasoning process is embedded in the user's prompt to guide the model's response (e.g., CoT), control lies with the user and we classify it as Structured Prompting; when reasoning is initiated and managed by an AI agent that plans, executes, and reflects over multiple steps, control lies with the system, and we classify it under Agent-based Orchestration.

5.3.3. In-Context Prompting

In-context prompting (Figure 14) leverages an LLM's ability to follow prompts that include a few examples (few-shot), a single example (one-shot), no examples but a task description (zero-shot), or instructions, to steer outputs toward desired behaviors [73,252]. (Note: Instruction Tuning is treated separately as a supervised method in §5.1). Its effectiveness hinges on prompt quality: well-designed prompts can emphasize factual accuracy and coherence, reducing hallucinations, yet prompt sensitivity and reliance on manual prompt crafting limit consistency across tasks and settings [73,252]. Within this paradigm, few-shot prompting supplies explicit input–output pairs in the prompt so the model infers task rules and patterns from context rather than fine-tuning; by grounding responses in these relevant examples, few-shot setups lower the risk of incorrect or hallucinated content [253]. More specifically, hallucination mitigation can be attributed to:

- Pattern Reinforcement: Exposure to multiple demonstrations helps the model align its response style and factual consistency with the provided examples. For instance, Principle-Driven Self-Alignment supplies 5 in-context exemplars alongside 16 human-written principles, giving clear patterns for compliance and thereby aligning the model's behavior with the desired norms [95].
- Bias Reduction: Balanced example selection can minimize systematic biases, particularly in ambiguous queries [254,255] while few-shot examples have been used to calibrate GPT-3's responses, demonstrating how different sets of balanced vs. biased prompts significantly influence downstream performance [71].
- Contextual Precision: The model learns implicit constraints from the given examples, preventing it from generating unrelated or misleading information [73,173].

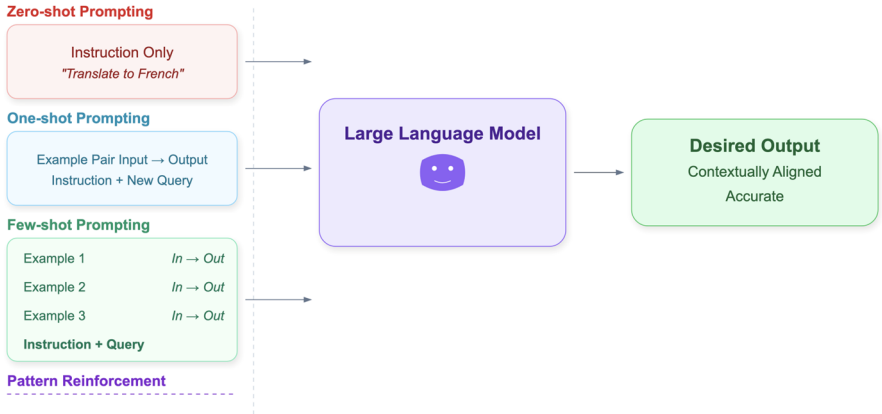

Figure 14. Illustration of in-context prompting strategies, including zero-shot, one-shot, and few-shot prompting.

Few-shot prompting is especially effective when paired with structured reasoning (e.g., CoT) to explicitly guide intermediate steps [20]. It can also support hallucination detection by prompting a model to flag ungrounded statements and route them to a separate mitigation agent for refinement [220,238]. In Chain of Verification (CoVe), few-shot exemplars enable self-correction by systematically checking outputs for factual errors [237]. Beyond this, few-shot examples can teach reasoning, enable multi-trajectory sampling with selection of the most consistent answer [21,62,242], demonstrate verification-and-revision workflows [256], guide negation-specific understanding [227], and probe hallucination tendencies of existing LLMs in a controlled manner [203].

Contrary to few-shot prompting, one-shot prompting provides a single example of the desired task format before the actual query, still aiming to lead the model to infer the expected response format [73]. Compared to zero-shot prompting, one-shot prompting significantly reduces hallucination rates by anchoring the model to at least one relevant reference point, but it may still be insufficient for complex reasoning tasks, particularly in domains requiring step-by-step logical inference or factual consistency [254].

Zero-shot prompting gives only a natural-language instruction—no demonstrations—and relies solely on pretrained knowledge to follow it [252]. It is computationally efficient and broadly applicable, but more prone to hallucinations due to the lack of contextual grounding [73,252]. Empirically, zero-shot baselines can fail; e.g., asking models to label statements true/false "completely failed," with accuracy $\leq 52\%$ [83]. To compensate, zero-shot setups often add explicit constraints (e.g., "cite sources," "answer only if certain") to promote factuality [9,53,203]. Several methods build on zero-shot prompting to curb hallucinations: Self-Familiarity uses two sequential prompts—first generating a concise explanation from internal knowledge, then reconstructing the concept from that explanation—to gauge fa-

miliarity and proactively prevent errors [84]; InterrogateLLM varies the prompt context to produce multiple answers, then detects hallucinations by spotting inconsistencies across responses [250]; and UPRISE automates natural-language prompts to improve zero-shot performance without manual prompt crafting [102].

5.3.4. Context Optimization

Context optimization supplies the model with the most relevant background, references, and constraints so it can produce accurate, grounded responses. By prepending carefully chosen passages or tokens before the query, it steers attention to salient details and lowers hallucination risk. Implementation can be external (e.g., retrieval-based selection [17,74]) or internal (e.g., meta-prompting [257], pruning redundant inputs to focus on key information [73], or contrastive decoding that amplifies context-aware probabilities [173]). It can also inject domain knowledge or curated datasets to condition the model on factual content, further reducing inaccuracies [219]. When token budgets are tight, methods like hierarchical compression [201] and sliding windows [258] preserve salient facts and prioritize grounded content.

Context optimization underpins RAG by selecting and fusing external evidence (e.g., from vector stores) into the prompt, thereby reducing reliance on parametric knowledge that is prone to hallucination [17,74,192]. Unlike conventional in-context prompting—which depends on prompt design alone—we treat RAG as an architectural modification (see §5.2) because it integrates a retrieval component directly into the inference pipeline. Beyond RAG, dynamic prompt construction spans critic-driven decoding [115], structured context tagging [219], and knowledge-aware conditioning [61]. For example, Ref. [195] enriches prompts with graph-derived signals, and [103] augments dialogue history and user queries with additional knowledge—both guiding models to prioritize relevant context and filter noise, improving factuality and accuracy. Finally, meta-prompting offers an advanced alternative to retrieval: it injects meta-level instructions into the prompt as task-agnostic, structured scaffolds for reasoning and verification, simplifying user interaction and enhancing generalization without external evidence [257].

5.3.5. System Prompt Design

While both prompt engineering and system prompt design influence LLM outputs, it is important to briefly differentiate between them since they operate at different levels of control and serve distinct functions. Prompt engineering focuses on optimizing a particular query's response through carefully structured user inputs that often involve instructions, constraints, examples, and continuous adjustments to phrasing in order to elicit a desired response [221,249], whereas system prompt design is used to create templates at the system level—invisible to the end-user—in order to establish behavioral and alignment guidelines that persist across multiple interactions and shape the model's default reasoning style, factual adherence, and response consistency [15,259].

System prompts can shape internal representations and attention, inject domain knowledge, and impose templates, scaffolds, and constraints—including ethical boundaries [81,133] and persona/tone settings that the model consistently inhabits [259]. Structured, constrained prompts that control sentence length, topical focus, and persona behavior steer ChatGPT-3.5 toward differentiated, domain-specific dialogues, enabling targeted mitigation development [260]. LaMDA similarly adapts via "domain grounding"/preconditioning, initializing with diverse snippets so it can assume application-specific personas [19]. Other work uses system-prompt design to inject hallucinated examples, reasoning instructions, rules, and few-shot demos to improve hallucination detection/mitigation (e.g., Llama3-70B) [113]. Knowledge Consistent Alignment (KCA) reconciles intrinsic

vs. external knowledge by appending reference snippets or shifting responses into refusal formats, training models to decline when uncertain [157]. In LinkQ, prompts guide users to refine questions and co-build queries with ground-truth KG data [209]; related prompts explicitly instruct verification, cross-referencing, and active correction of misinformation [261]. Finally, evolving-instruction methods expand prompts in-depth (adding constraints or reasoning steps) or in-breadth (mutating instructions) and filter invalid instructions, collectively improving complex instruction following [123].

Beyond explicit prompt engineering, LLMs appear to internalize personas—distinct behavioral characteristics learned from pretraining data—such that the truthfulness of an answer can often be predicted from pre-generation activations [259]. Using contrastive prompting and activation differencing, researchers extract persona vectors—linear directions in activation space tied to traits like hallucination propensity or sycophancy—enabling pre-finetuning data screening and post hoc control by steering these vectors toward ethical/behavioral guidelines [262]. However, persona drift challenges the persistence of system prompts: over lengthy dialogues, models can lose the assigned persona or even adopt user traits, suggesting prompt influence decays and alignment can destabilize via attention dynamics [243]. Thus, relying solely on system prompts for stable persona alignment is unreliable; system prompts are inherently broad and often need more granular prompt engineering or RLHF. For example, Ref. [15] employs instruction-style prompts to encode behavioral expectations and then fine-tunes with a combination of supervised learning and RLHF, which—though not persistent system messages—function similarly by shaping global behavior during training.

Beyond textual system prompts at inference, prompt tuning [235], prefix tuning [263], and postfix tuning [253] attach learned continuous vectors to steer behavior: prompt tuning appends vectors to the system message in the first layer, prefix tuning inserts them in all layers before the system message, and postfix tuning places them in all layers after it. Though not designed specifically for hallucination mitigation, these soft prompts/internal prefixes are increasingly used to control model behavior, including reducing hallucinations, without modifying base weights. A key limitation is interpretability: unlike human-readable prompts or meta-prompts, the learned embeddings are opaque, making it difficult to audit or understand how they influence hallucination behavior [253,263].

5.4. Post-Generation Quality Control

Post Generation Quality Control encompasses a set of post-generation checks applied to text outputs, aiming to identify or correct inaccuracies after an intermediate or final text has been generated. Among these methodologies are:

- Self-verification and Consistency Checking: Involves internal assessments of output quality, ensuring logical flow, and maintaining factual coherence within the generated content.
- External Fact-checking and Source Attribution: Validates information against outside authoritative sources or asks the model to explicitly name its sources.
- Reliability Quantification: a broader subcategory that encompasses:
  - Uncertainty Estimation (quantifying the likelihood of claims) and
  - Confidence Scoring (assigning an overall reliability score to the output).
- Output Refinement: Involves further shaping and iteratively polishing the generated text.
- Response Validation: Strictly focuses on confirming that the output meets specific, pre-defined criteria and constraints.

5.4.1. Self-Verification and Consistency Checking

Self-verification (Figure 15) refers to the model's introspective evaluation of a single output's correctness using its own internal knowledge and reasoning. This includes re-parsing, self-questioning, or generating challenges to assess the factuality or coherence of its own response. Unlike consistency checking, which compares multiple outputs, self-verification focuses on whether the model can internally judge and refine a specific answer. While conceptually distinct, self-verification often overlaps with consistency-based methods, as both aim to detect hallucinations through internal evaluation.

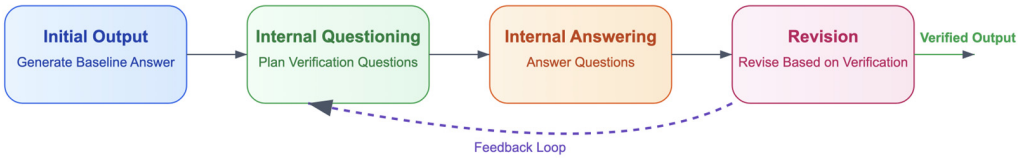

Figure 15. Simplified diagram of Self-verification.

A common approach is multi-step introspection, as in CoVe [237], where the model produces a baseline answer, plans targeted verification questions, answers them, and revises its response—without external tools. Related iterative self-checking frames self-verification as a loop of factual evaluation and internal correction [45], while Socratic prompting employs elenchus/dialectic to guide multi-turn internal Q&A that refines reasoning and exposes inconsistencies [249]. Complementary work shows introspective error recognition, with models identifying their own incorrect claims in explanations [65], and reference-focused self-verification by directly querying whether the model has sufficient, consistent internal information about cited references [4]. Further, Ref. [264] studies whether LLMs can self-verify by introducing P(True) (probability the model's sampled answer is correct) and P(IK) (whether the model "knows" the answer pre-generation), alongside calibration analysis and self-evaluation prompts. Finally, DrHall [265] detects and reduces factual hallucinations via metamorphic testing: it rewrites queries into semantically equivalent forms to induce different internal execution paths, checks answer consistency, and corrects errors through multi-path voting.

Several works embed self-verification directly into the model or training pipeline. Self-Checks builds verification into the architecture via rephrasing, iterative refinement, and cross-checking of generated outputs [266], while the SLM mechanism explicitly learns self-evaluation from an LLM to assess its own outputs [153]. A dual-model setup evaluates one model's output with another using token-level confidence, embedding coherence, and probabilistic anomaly detection—without human/external feedback [213]. Complementary implicit feedback schemes include a corrector model that refines a generator via feedback loops aligned to task constraints/signals (e.g., toxicity, functional correctness) [23], and two decoding-time techniques—Self-Diagnosis (the model flags biases in its own output) and Self-Debiasing (adjusts decoding to reduce biased content without weight updates) [80]. Finally, marginalization operationalizes self-verification by sampling diverse reasoning paths and selecting the answer most consistently supported across these self-generated derivations, reflecting the intuition that complex problems have a single solution reachable via multiple paths [21].

Consistency checking evaluates whether a model's outputs remain factually and logically stable across repeated or reformulated queries, emphasizing behavioral stability over time/phrasing—unlike self-verification, which inspects a single output—though many methods blend both [220,228,250,267]. A common tactic is to surface self-contradictions across multiple generations: AutoHall prompts the model to produce independent references for a claim and compares them to the original; contradictions signal a flawed

understanding and potential hallucination [228]. InterrogateLLM queries the model with interrogation-style prompts and flagging inconsistencies across its own answers as hallucination indicators [250]. SelfCheckGPT elicits multiple answers to the same input and measures semantic variability among them; higher divergence serves as a proxy for uncertainty or hallucinations [267]. The CoNLI framework provides a more structured pipeline where each sentence is treated as a hypothesis and checked against the source followed by NER to ensure all entities trace to the source [220].

Repeated prompting can test behavioral consistency. In [117], after a potentially false statement, the model answers a sequence of unrelated follow-up questions; post hoc analysis of the yes/no responses detects distinctive lie-related behavioral patterns. Similarly, Ref. [11] examines reliability under identical conditions in military crisis simulations by prompting the same scenario multiple times and using BERTScore to quantify semantic differences across responses.

Finally, several methods at the boundary of self-verification and external fact-checking warrant clarification due to their hybrid nature. Specifically, CRITIC [256] performs an introspective reasoning loop but relies on external tools for evidence. DreamCatcher [135] primarily serves as an external verifier by comparing model outputs to Wikipedia references, although it also probes the model's internal knowledge. Additionally, Knowledge Consistent Alignment (KCA) [157] uses alignment data containing verified facts to detect inconsistencies, functioning as an external check while RefChecker [230] explicitly operates in two modes: a zero-context setting for pure self-verification and a Noisy Context mode for external verification using retrieved knowledge. Finally, FLEEK [207] and KGR [61] exemplify hybrid pipelines where factual claims are first extracted from outputs and then validated against knowledge graphs, with the retrieved evidence sometimes feeding into a secondary generation loop.

5.4.2. External Fact-Checking

External fact-checking (Figure 16) is a post hoc validation step that verifies model outputs against third-party or curated sources. Typically automated via APIs, systems search authoritative databases/websites [57,110,116,127,244,246], consult knowledge graphs [61,188,204,209], or pull from code/source repositories [66] to compare claims with documented facts and assign veracity or confidence scores. By anchoring responses to external evidence—less susceptible to training idiosyncrasies—these checks can mitigate hallucinations. However, effectiveness depends on the availability and reliability of external sources; borderline or contradictory results often require a human-in-the-loop, and grounding still does not guarantee absolute accuracy [9,46]. As such, external fact-checking is best viewed as a complement to internal methods like self-verification, helping align outputs with established records. While some hybrid approaches (e.g., FLEEK [207], KGR [61], LinkQ [209]) reintegrate retrieved facts into a secondary generation loop, most methods operate post hoc and leave the model architecture unchanged—in contrast with Section 5.2.4 "Knowledge Representation Approaches," which modify inference or latent representations to inject external knowledge directly.

Beyond post hoc checks, many methods integrate external knowledge directly into the LLM's generation or revision loop. ANAH validates sentences by retrieving references from Wikipedia and Britannica [57]. LLM-AUGMENTER injects web or task-specific evidence into prompts, with a Knowledge Consolidator for retrieval, entity linking, and evidence chaining [193]. DRAD triggers a Self-correction via External Knowledge (SEK) module to fetch evidence when hallucinations are detected and revise outputs [55]. Verify-and-Edit introduces external knowledge from DrQA/Google Search to correct reasoning chains [268]. LaMDA employs a "Research" phase that iteratively queries sources and integrates findings

during dialogue generation [19]. PURR performs external fact-checking by denoising LM corruptions with retrieved evidence to fix factual errors [97]. REFCHECKER evaluates claims against retrieved knowledge [230]. ProbTree uses open-book QA to compare internal vs. retrieved evidence, selecting the more confident answer [198]. HERMAN verifies quantity entities (e.g., dates, numbers) in summaries against the source article to ensure factual consistency [269].

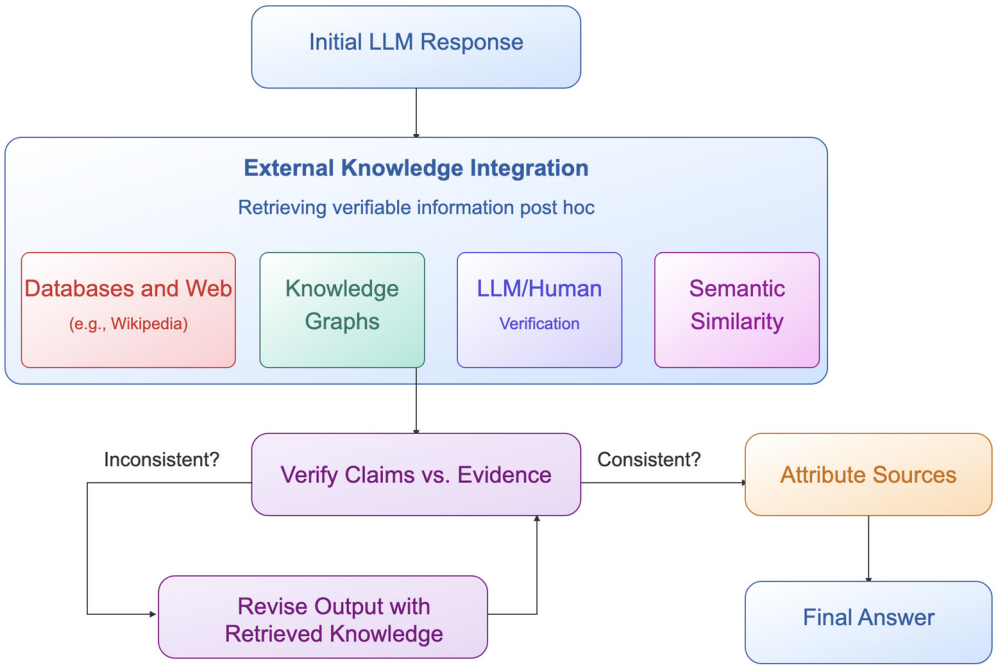

Figure 16. Flow diagram of the External Fact-Checking process, showing verification stages, intermediate validation, and feedback loops for iterative refinement of model outputs.

Several frameworks inject knowledge graphs (KGs) as external ground truth to verify and refine LLM outputs. KGR [61] extracts claims, identifies entities, retrieves relevant triples, and verifies statements against KG facts. LinkQ [209] compels LLMs to construct and execute KG queries, ensuring answers come from up-to-date KG data. Neural Path Hunter [188] follows a generate-then-refine loop, using a token-level hallucination critic to amend responses so entities/relations are KG-supported. Drowzee [245] grounds outputs in a curated symbolic KB (e.g., from Wikidata) and uses backward chaining plus abductive reasoning over entity–relation–entity triples to enforce consistency. Chain of Knowledge (CoK) [270] targets hallucinated rationales by injecting grounding via SPARQL over heterogeneous sources. Finally, Ref. [110] uses model-based extraction of relation tuples from generated text, verifies them against ground-truth summaries, constructs a large-scale dataset by cross-referencing Wikidata, and defines a factual accuracy (factacc) metric (precision over extracted tuples) to quantify correctness.

Several works enlist LLMs—often with human oversight—for external fact-checking. FactScore retrieves Wikipedia evidence and uses a fact-verification model to check atomic claims [127]. In [232], GPT-4 retrieves context and acts as an independent reference; entailment models compare answers against this context to flag contradictions. Ref. [5] similarly has GPT-4 extract standalone factual statements from outputs and contrast them with its rich world knowledge, mimicking external verification. TrueTeacher has an LLM annotate summaries for factual consistency against source documents [98]. Self-Checker employs AMT human annotators for claim detection, evidence retrieval, and veracity prediction on BINGCHECK [22], and [224] uses human labelers (often with search/Wikipedia) to

assess ChatGPT outputs. By contrast, EVER automates verification with an NLI-based verifier over Wikipedia passages; the resulting factual feedback trains a reward model that steers generation via RL, improving consistency with external sources [246]. FacTool decomposes answers into atomic claims, retrieves evidence (e.g., Wikipedia), and applies a fact-verification model to each claim [244]. Finally, MIXALIGN maps user constraints to knowledge bases; if mapping is uncertain, it issues follow-up questions to refine candidate groundings before generating the final answer [205].

Beyond injecting knowledge from KGs or other LLMs, many works use consistency and semantic similarity for external fact-checking. Ref. [271] cross-references LLM outputs with structured KBs, combining anomaly detection and coherence analysis to flag factual deviations. For translations, Ref. [272] measures source–hypothesis match via cross-lingual semantic similarity (LaBSE, LASER), NLI (XNLI), and quality estimation (COMET-QE). DreamCatcher [135] compares generations to Wikipedia with token overlap and cosine similarity, also probing internal knowledge via activations. Ref. [116] improves open-ended generation by checking against Wikipedia using NEER (Named Entity Error Rate) and EntailR (entailment ratio). FLEEK [207] extracts fact triples, generates type-aware questions, retrieves evidence (KGs + web), and judges claims by comparing original vs. retrieved triples. HDM-2 [111] leverages parametric "common knowledge" to spot contradictions and adds span-level verification against retrieved context. Ref. [104] injects KB-derived triplets/summaries and applies entailment for response consistency. CRITIC [256] interfaces with Google Search, Python, and toxicity APIs; Ref. [66] retrieves code snippets from a local repo (similarity-based) as ground-truth evidence. Ref. [273] introduces Hierarchical Semantic Pieces (HSP) to extract multi-granularity semantics from outputs and references, then uses a fact verifier to score consistency. Finally, KCA [157] employs external snippets to create multiple-choice questions, testing and aligning model outputs with referenced knowledge.

Source attribution identifies where a response's information comes from—training data, external datasets, or retrieved snippets—mitigating hallucinations by steering models toward verifiable evidence and increasing accountability by tying claims to records. HAR (Hallucination Augmented Recitations) uses counterfactual open-book QA datasets so answers must be grounded in the provided text; models fine-tuned on HAR are explicitly rewarded for attributing to the document rather than parametric memory [42]. RARR performs "Research & Revision" by consulting an external Document Corpus to post-edit outputs, revise unsupported claims, and produce an attribution report [190,274]. Ongoing challenges include handling partial matches, aggregation across multiple sources, and contradictory evidence in both settings.

5.4.3. Uncertainty Estimation and Confidence Scoring

In this category, we examine two post hoc approaches for quantifying the reliability of LLM outputs: uncertainty estimation and confidence scoring. Uncertainty estimation assesses how sure the model is by probing reliability and consistency across different scenarios or model states, often leveraging architectural adjustments or prompt engineering to elicit uncertainty signals. By contrast, confidence scoring produces a single numerical value for a given output, indicating how confident the model is in that specific response. While both measure "sureness," they differ in scope and use: uncertainty estimation is a more complex, probabilistic view suited to analysis and risk-aware decisions, whereas confidence scoring offers a simple, operational signal appropriate for system-level thresholds and routing.

Uncertainty Estimation

In LLMs, uncertainty spans aleatoric (inherent ambiguity or multiple valid interpretations) and epistemic (model/knowledge limitations that can be reduced with more or better data) [68]. Uncertainty estimation elicits how unsure the model is—typically via Monte Carlo dropout, Bayesian approximations, or calibration layers—yielding a distribution over outputs or a range of plausible alternatives rather than a single score. These signals enable systems to flag risky responses for external review (e.g., by a user or verifier). In practice, many approaches are hybrid, combining architectural modifications or prompt-engineering probes to capture the model's true state of doubt.

- Entropy-based approaches: The Real-time Hallucination Detection (RHD) flags high-entropy entities and triggers self-correction when unreliability is predicted [55]. Conditional Pointwise Mutual Information (CPMI) quantifies token-level conditional entropy, identifying hallucinated tokens as high-entropy states and reinforcing uncertainty as a useful proxy [175]. INSIDE computes an EigenScore—differential entropy in the sentence-embedding space—directly from hidden states and applies feature clipping to curb overconfident generations [26]. A PMI-based detector measures "overshadowing" via perturbations, using uncertainty as a cue to spot low-confidence conditions [54]. Beyond surface entropy, semantic formulations better capture meaning-level doubt. "Semantic entropy" clusters multiple sampled answers by meaning and computes entropy over clusters, estimating uncertainty over interpretations rather than word choices [25]. Semantic Entropy Probes (SEPs) infer a comparable signal from a single generation's hidden states—eschewing multi-sample costs and outperforming log-probability or entropy baselines [275]. Hybrid estimators further improve reliability by pairing auxiliary models with decoding signals: an Epistemic Neural Network (ENN) extracts hidden features, trains a small MLP for next-token prediction, and fuses its outputs with contrastive-decoding logits to down-weight low-confidence generations [276]. CHOKE shows high-certainty hallucinations—even when the model "knows" the answer—across semantic entropy and token probability [79]. Hence, these signals are valuable early-warning indicators but should be paired with external verification or contrastive/decoding controls to detect high-confidence failures.
- Sampling: Sampling-based uncertainty estimation treats variability across multiple generations as a proxy for uncertainty. In [127], the model is sampled repeatedly and output divergence is quantified to produce a resampling-derived confidence value (distinct from single-token probabilities); this scalar then serves both as a verification signal and as a reward during reinforcement learning. Similarly, Ref. [267] detects hallucination risk by measuring divergence among sampled responses, operationalized with metrics such as BERTScore and NLI-based agreement. Extending this idea, Ref. [77] three main components: sampling strategies to generate multiple responses, different prompting techniques to elicit the model's uncertainty and aggregation techniques which are used to combine these multiple responses and their associated confidence scores to produce a final, calibrated confidence score.
- Monte Carlo methods: Fundamental measures such as sequence log-probability and Monte-Carlo dropout dissimilarity are used as uncertainty signals to detect hallucinations and to drive downstream refinement, detection, and re-ranking, capturing variability and confidence in predictions [9]. In [233], reward estimation based on the log probability of actions effectively quantifies confidence in individual reasoning steps; MCTS then exploits these rewards to prioritize higher-plausibility paths. Although not labeled as "uncertainty estimation," this setup substantially overlaps with it, since the reward function encodes a trust/uncertainty signal over the LLM's reasoning traces [233].

- Explicit Verbalization: The core method in [155] trains models to verbalize epistemic uncertainty by distinguishing "uncertain" vs. "certain" data using both supervised and unsupervised signals, and evaluates calibration with ECE and AP, improving models' ability to express self-doubt. SelfAware introduces a benchmark and method for detecting when a model should state uncertainty in response to unanswerable questions [277]. Similarly, Ref. [278] fine-tunes GPT-3 to output "verbalized probability," a direct expression of epistemic uncertainty— a higher-order objective beyond raw softmax confidence. While we place [278] under Uncertainty Estimation, confidence scoring remains crucial: calibration is assessed with MSE and MAD, but these scores arise from the verbalized probabilities themselves, not from post hoc logits.
- Semantic analysis: In [78], semantic density quantifies uncertainty by measuring the similarity between a given response and multiple completions in embedding space. It operates response-wise (not prompt-wise), requires no retraining, and addresses limitations of earlier uncertainty methods (e.g., semantic entropy, P(True)). Although the authors consistently describe the resulting scalar as "confidence," the [0, 1] score—with natural thresholds for filtering—is best viewed as the output of an uncertainty metric. In [232], semantic analysis is combined with logit-derived, token-level probabilities to compute a confidence score per atomic unit, which is then integrated with textual entailment probabilities to yield a refined score for detecting hallucinated spans. While termed "confidence" and useful for thresholding, this score functions within a broader hallucination-detection pipeline; thus, confidence scoring is effectively a means to uncertainty estimation in the authors' framework.
- Training approaches: [152] explicitly links hard labels to model overconfidence and proposes soft labels to introduce uncertainty-aware supervision. By restructuring the training objective to reflect confidence calibration and evaluating overconfidence via the NLL of incorrect answers, the authors show that fine-tuning with soft labels reduces misplaced certainty—an important driver of hallucinations. Ref. [36] extends this idea by using smoothed soft labels (rather than hard labels) to mitigate hallucinations through knowledge distillation: a student model learns from a calibrated probability distribution, consistent with the maximum-entropy principle, yielding better factual grounding and reliability. As in [152], overconfidence is assessed by plotting NLL on incorrect answers, and soft-label fine-tuning is shown to reduce unwarranted certainty.
- Composite methods: In [175], epistemic uncertainty directly guides decoding via a modified beam search that prioritizes low-uncertainty continuations, reducing incorrect or nonexistent facts. Ref. [279] uses a proxy model to compute token- and sentence-level hallucination scores from uncertainty metrics; these signals are sharpened by (i) emphasizing keywords, (ii) propagating uncertainty through attention weights, and (iii) correcting token probabilities by entity type and frequency to address over- and under-confidence. Finally, in [168] the authors use the attention mechanism as a self-knowledge probe. Specifically, they design an uncertainty estimation head, which is essentially a lightweight attention head that relies on attention-derived features such as token-to-token attention maps and lookback ratios, serving as indicators of hallucination likelihood.

Confidence Scoring

Confidence scoring assigns a single numerical value reflecting the model's conviction in the correctness of each token, phrase, or full response. Typically derived from the model's underlying softmax probabilities, these scores provide a more lightweight and actionable measure. The practical impact is that it allows the deployment system to implement threshold-based decisions—such as refusing to provide an answer below a

certain confidence level or requesting human oversight for borderline outputs. While confidence and uncertainty are two sides of the same coin, confidence scoring focuses on the practical application of a single, calibrated score for system-level decision-making, in contrast to the more intricate and probabilistic nature of uncertainty estimation.

Confidence signals are used both to score and steer generation. In [213], token-level confidence scoring assigns a value to each token; low scores flag likely hallucinations and feed a feedback loop that dynamically revises outputs. HILL operationalizes confidence for users—exposing Colored/Ordinal/Metric confidence scores and a confidence-threshold slider to gate responses [225]. Probabilistic Tree-of-Thought reasons over a query tree while tracking confidence for both decomposition and answering, computing log-likelihood over explanation sequences to pick the most reliable answer [177]; Similarly, Ref. [198] uses the same log-likelihood criterion to choose between retrieved vs. parametric sources of truth. Self-Highlighted Hesitation (SH^2) treats low-confidence tokens as informative: it appends them to induce "hesitation," then scales prediction probabilities based on the confidence gap between original and hesitated inputs, nudging the model toward factual content [179]. Finally, Ref. [261] analyzes confidence shifts in multi-turn dialogue, using token probabilities to show how persuasive misinformation can distort a model's confidence distribution. Collectively, these methods elevate confidence from a passive metric to an active control signal for detection, selection, and real-time refinement.

Finally, in [76], the authors propose a confidence score, which is used to detect hallucination by measuring how much the model attends to the source and how much a word conveys source information. Specifically, it introduces a token-level confidence score, computed by combining an attention-based score with the probability from a base language model. This confidence score is used both during training—via a variational Bayes framework—and at inference, via reranking [76].

5.4.4. Output Refinement

Output refinement refers to post-processing methods that improve an LLM's initial response before presentation. Approaches fall into two broad families. The first, external-evidence methods, improves outputs by using information from outside the model's original training data. This includes techniques like retrieval-augmented generation (RAG), which uses web searches or other external sources, as well as reasoning over structured data from sources like knowledge graphs (KGs). The second family, internal self-improvement methods, relies on the model's own capabilities to refine its output, such as through iterative self-correction or using its own self-generated data.

Methods that leverage external evidence encompass a set of approaches where LLMs collect information beyond their initial training data to refine and ensure the factual accuracy or coherence of their outputs. There exist a number of papers employing various such methods:

- RAG/Web-search–based output refinement retrieves external evidence to correct or revise model outputs. CRAG uses a lightweight retrieval evaluator and a decompose–then–recompose strategy to judge the relevance of retrieved documents for a given query [280]. EVER validates generations against verified sources and iteratively fixes intrinsic errors or reformulates extrinsic hallucinations [246]. FAVA conducts span-level hallucination detection and editing, marking inaccurate or subjective spans for deletion and proposing corrected replacements to refine the final text [93].
- Structured knowledge sources integrate and reason over formal data—e.g., knowledge graphs (KGs)—to refine and validate LLM outputs. Ref. [167] couples text with relational signals via GNN-based probabilistic inference to improve factual grounding, while [183] re-ranks conversational candidates using walks over KG subgraphs to en-

hance reasoning. Neural Path Hunter (NPH) [188] follows a generate-then-refine loop, detecting hallucinated entities in dialogue and replacing them via targeted KG queries. During FLEEK's revision phase [207], a fact-revision module proposes corrections to dubious triples using verified KG/Web evidence. Complementing graph-centric methods, Ref. [82] blends deductive formal verification with the inductive strengths of LLMs, using logical tests to expose hallucinations.

- External Feedback and verification: These methods refine outputs using outside signals—human feedback, retrieved evidence, or verified knowledge. CRITIC [256] exploits CoT and few-shot prompting to revise hallucinated answers based on external feedback (free-form QA, mathematical reasoning, toxicity reduction). Chain of Knowledge (CoK) [270] applies a three-stage pipeline—reasoning preparation, dynamic knowledge adapting, and answer consolidation—and, when no majority consensus emerges, corrects rationales by integrating heterogeneous sources. Verify-and-Edit [268] specifically post-edits CoT chains: the model produces an initial answer and CoT retrieves external knowledge to answer them. Subsequently, it adjusts the original CoT and final response to fix unsupported or incorrect claims. Within DRAD [55], the SEK module acts when RHD flags a likely hallucination: SEK formulates a query from the local context, retrieves relevant evidence, truncates the output at the error point, and regenerates the continuation using the retrieved knowledge.
- Filtering Based on External Grounding: These methods filter outputs by checking them against external documents or ground truth. HAR [42] uses Factuality Filtering and Attribution Filtering to retain only answers explicitly supported by the provided document. HaluEval-Wild [107] applies filtering, manual verification of hallucination-prone queries, and selection of difficult cases to refine outputs and ensure the evaluation set contains only well-grounded examples.
- Agent-Based Interaction with External Context: These involve agents that interact with external environments or receive structured external feedback for refinement. For instance, the mitigation agent in [220] is designed to refine and improve the output by interpreting an Open Voice Network (OVON) JSON message. This JSON message contains crucial information, including the estimated hallucination level and detailed reasons for potential hallucinations, which guides the refinement process [139].
- Model Tuning/Refinement with External Knowledge: Methods that explicitly use external knowledge during their training or refinement phase to improve model outputs. In [157], methods like refusal tuning, open-book tuning, and discard tuning are leveraged to refine the outputs of the model, thus ensuring consistency with external and intrinsic knowledge. The PURR model refines its outputs through a process akin to conditional denoising by learning to correct faux hallucinations—intentionally corrupted text that has been used to fine tune an LLM. The refinement happens as PURR denoises these corruptions by incorporating relevant evidence, resulting in more accurate and attributable outputs [97].

Complementary to methods that leverage external information, self-improvement methods represent a distinct paradigm where LLMs leverage their own internal capabilities or self-assessment mechanisms to iteratively refine model outputs, often reducing or completely eliminating reliance on external feedback or extensive manual annotation. There exist a number of papers employing various self-improvement techniques:

- Iterative self-correction methods refine outputs through repeated, prompt-driven revision and internal checks. An adaptive framework in [247] performs defect analysis, guided optimization, and response comparison via prompt-based voting. Self-Checks [266] rephrase prompts or ask related questions to test internal consistency, while [281] uses in-context prompting to incorporate the model's self-generated feed-

back for iterative correction. Self-Reflection [45] rewrites answers to improve factuality, consistency, and entailment, and [67] prompts the model to identify and adjust self-contradictions within its own text. Finally, Tree of Thoughts (ToT) [234] employs structured search to explore and evaluate intermediate reasoning branches, enabling staged self-evaluation and refinement of the reasoning path.

- Self-Regulation during Generation/Decoding, where the model re-adjusts its own output or decision-making process in real-time during generation. For instance, the Self-highlighted Hesitation method (SH2) presented in [179] refines the model's output by iteratively recalibrating the token probabilities through hesitation and contrastive decoding, while the Hypothesis Verification Model (HVM) estimates faithfulness scores during decoding, refining the output at each step [119].
- Self-Generated Data for Improvement, where the LLM generates data or instructions which are subsequently used to finetune itself. For instance, the Self-Instruct framework bootstraps off the LLM's own generations to create a diverse set of instructions for finetuning while in WizardLM such instructions are evolved and iteratively refined through elimination evolving to ensure a diverse dataset for instruction fine-tuning.
- Model-based techniques and tuning: Approaches in this group refine outputs by adding evaluators, rerankers, or specialized training. LaMDA uses a generate-then-rerank pipeline with discriminators that score safety and quality, selecting the top candidate [19]. Dehallucinator overwrites flagged translations by sampling Monte Carlo-dropout hypotheses, scoring them, and choosing the best translation [282]. A dual-model setup [213] pairs a generator with an evaluator that applies token-level confidence scoring and probabilistic anomaly detection; a feedback loop flags problematic spans and iteratively adjusts the output. SC2 (Structured Comparative reasoning) combines approximate inference with pairwise comparison to pick the most consistent structured representation from multiple intermediates [242]. Ref. [77] improves reliability by sampling multiple responses and aggregating them for consistency. Verbose Cloning [95] uses tailored prompts and context distillation to make answers more comprehensive, reducing overly brief or indirect outputs. A corrector model [23] iteratively upgrades a base model's hypotheses via value-improving triplets (input, hypothesis, correction), yielding gains in math program synthesis, lexically constrained generation, and toxicity removal. Finally, an MoE architecture [212] refines outputs through expert consensus—using majority voting to filter erroneous responses and retain only agreement-backed generations.

5.4.5. Response Validation

Response validation evaluates the final or near-final output against specific, predefined criteria and constraints, either automatically through rule-based systems or semantically through learned models. This process may include checking for factual correctness [225], verifying mathematical reasoning [225], or matching against known templates in specialized fields. More advanced validators capable of parsing sentence-level logical coherence have also been proposed [232,273]. Given its nature, response validation can be easily updated or expanded as new types of errors are discovered, thus providing a structured safety net, catching errors too subtle for earlier training or architectural methods.

In neural machine translation, a detect-then-rewrite pipeline [272] samples multiple hypotheses via Monte Carlo dropout, then validates each with attribution/similarity criteria to select the most valid output. In [282], hallucinations are defined as translations unfaithful to the source. Dehallucinator flags suspect cases and scores them with COMET-QE, ensuring adherence to source faithfulness. For abstractive summarization, HERMAN selects the best summary by verifying that quantities (dates, numbers) in candidates match

the source [269]. HILL is an external aid that surfaces confidence scores, sources, disclosure, and visual cues, running validation checks to help users judge factuality and hallucination risk [225]. In code generation, CRITIC separates validation from correction: it checks logical validity via execution and factuality via search, then applies edits accordingly [256].

Fine-grained response validation targets step- or span-level checking of outputs. In mathematical reasoning, FG-PRM ranks candidate solutions and assigns process-level reward scores to confirm whether each reasoning step satisfies predefined mathematical-correctness criteria [113]. Hierarchical Semantic Piece (HSP) detects and corrects hallucinated segments by (i) computing semantic similarity between sentence-level pieces of the generation and corresponding reference text and flagging pieces below a threshold, and (ii) performing KB lookups over entities/relations extracted from both the output and the reference; flagged spans are then paraphrased, rewritten, or removed to mitigate errors [273]. Complementing these, Ref. [232] decomposes answers into atomic units and evaluates each unit's alignment with retrieved context by combining textual entailment probabilities with token-level confidence into a refined score indicating hallucination likelihood; units falling below a set threshold are classified as hallucinations and targeted for correction. Collectively, these methods operationalize validation at the granularity of steps or spans, enabling precise detection and localized revision rather than coarse, output-level judgments.

5.5. Interpretability and Diagnostic Approaches

Interpretability and diagnostic approaches primarily focus on detection methods that help researchers understand why and where a model may be hallucinating. These include techniques such as internal state probing (which examines the model's internal variables and hidden representations), attribution-based diagnostics (which link outputs to specific inputs or internal steps), and neuron activation and layer analysis (which investigates activity in particular model components). These strategies—often collectively referred to as mechanistic interpretability approaches—aim to reverse-engineer deep neural networks and guide refinements in training or architecture as well as provide insights into model behavior based on its internal representations. Collectively, these approaches demystify hallucination triggers—whether through learned probes, activation patterns, or input attributions—thus providing a foundation for architectural refinements.

5.5.1. Internal State Probing

Internal state probing (Figure 17) analyzes hidden representations with diagnostic classifiers (probes) to reveal what information a model encodes and how inputs propagate to outputs [283]. Studies show systematic links between internal states and hallucinations: probing can surface linguistic features or hallucinations [185,271], assess response consistency [104], and detect unlabeled anomalies [284]. It also helps evaluate alignment between internal mechanisms and behavior, using tools like PCA and linear classifiers to illuminate otherwise opaque decision processes [112]. However, probes have limitations—they can overfit synthetic data, fail to generalize across tasks, or even introduce artifacts rather than expose true model knowledge [194,226].

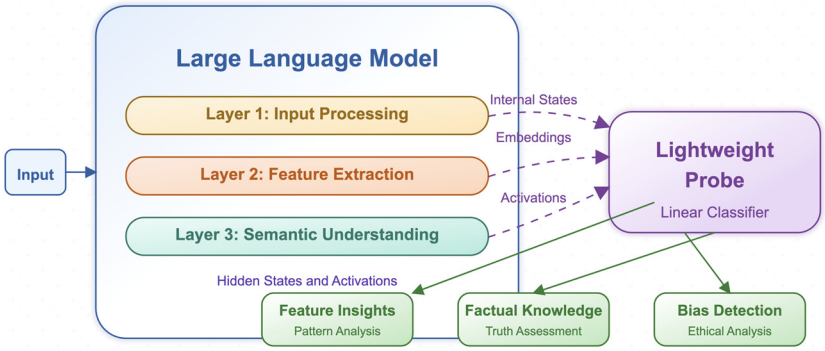

Figure 17. Visualization of internal state probing used to analyze hidden layer representations.

Although internal state probing can employ learned components, its core aim is interpretive diagnosis of a model's internal representations. In our taxonomy, we assign papers to Internal State Probing when their primary contribution is a probing classifier; otherwise, we place them under Neuron Activation and Layer Analysis or Attribution-based Diagnostics, depending on the dominant method. We categorize by the primary diagnostic approach, while acknowledging hybrids—e.g., Ref. [148] blends probing with gradient-based attribution—and note that some works span multiple techniques even if we list them under a single heading.

In [271], the internal states of OpenAI's ChatGPT and Google's Gemini are compared for hallucination susceptibility by crafting/refining hallucination-inducing prompts across diverse topics, then extracting attention-weight patterns and hidden-layer activations. Regression analysis and PCA reveal strong correlations between specific internal-state parameters and hallucination frequency; practically, the authors propose detection via anomaly detection plus linguistic analysis and mitigation via cross-referencing with structured knowledge bases. Complementarily, Ref. [194] asks whether the diagnostic signal needed to detect hallucinations already resides in self-attention output states or feed-forward layers. The authors train probing classifiers of increasing complexity across abstractive summarization, knowledge-grounded dialogue, and data-to-text generation to detect span-level hallucinations in both sampled and synthetic datasets, and conclude that narrowly trained probes generalize poorly across tasks and behave inconsistently between natural and synthetic hallucinations.

Extending [170,194] treats factual queries as constraint criteria and introduces a SAT-probe classifier, finding a strong positive correlation between attention to constraint tokens and factual accuracy; concentrating probe attention solely on these tokens can match or exceed next-token likelihood maximization, though attention alone cannot explain all failures. Building on interpretable activation directions, Inference-Time Intervention (ITI) [185] uses linear/orthogonal probing to identify a small set of high-accuracy attention heads and manipulates their activations along truth-correlated directions to reduce hallucinations. Nonlinear ITI (NL-ITI) [166] generalizes this with nonlinear probes and a multi-token intervention strategy, extracting richer internal signals and steering generation toward more truthful outputs. Lookback Lens [285] computes a lookback ratio—the share of attention to input-context tokens versus newly generated tokens across layers/heads—and trains a logistic regression classifier on these features to detect hallucinated spans.

INSIDE shifts uncertainty estimation from tokens to sentences by analyzing dense semantic information in internal states. It computes EigenScore—the leading eigenvalues of the covariance matrix over sentence-embedding sets from multiple generations—to quantify semantic dispersion; higher scores indicate greater diversity and potential hallucination, which is then countered with feature clipping to curb overconfident responses [26]. Semantic Entropy Probes (SEPs) train a linear logistic regressor on hidden states paired with

semantic-entropy targets, yielding a computationally efficient proxy for faithfulness [78]. Probing studies further show that truthfulness can be predicted pre-generation from internal states; finetuning on factual QA pairs strengthens alignment with a "truthful persona" and improves out-of-domain truthfulness [286]. Mechanistic analyses manipulate residual-stream activations to locate a "truth" signal: token-and-layer patching reveals that inserting residuals from a true prompt into a false one can flip the model's TRUE/FALSE logits [287]. Complementarily, other work shows hidden states carry linearly separable "lying" signals independent of final logits—i.e., the model can internally "know" it is lying even when the surface output asserts otherwise—exposing a dissociation between latent knowledge and generated text [83].

LLM Factoscope decomposes outputs into discrete factual units and locates their associated "fact clusters" in hidden representations; a small linear probe then tests whether each fact is supported by the model's internal states, distinguishing internally encoded from fabricated claims [148]. Building on related evidence that internal "truth" directions exist [30,148,287], TruthForest detects such truth representations and—unlike those works—actively intervenes: orthogonal probes steer activations along truth-aligned directions, and Random Peek applies this intervention across many sequence positions to bias attention toward truthful information [120,121]. PGFES proposes a two-stage mitigation pipeline: attention-augmented MLP probes identify fine-grained, type-specific "truthfulness directions" in intermediate activations; at inference, the model edits hidden states along these directions to generate multiple low-risk candidates. Subsequently, it uses dynamic, similarity-weighted sampling to assemble the most semantically consistent final response [288]. Finally, Lightweight Query Checkpoint (LQC) extracts intermediate hidden states from a smaller LLM, forms query embeddings, and trains a binary classifier to decide if a query requires verification [289].

Beyond locating truthful statements, internal state probing can test whether LLMs encode beliefs by extracting hidden-layer embeddings and training a separate probe to infer those beliefs without the original input text [226]. Complementary work shows that latent representations inherently encode (un)answerability, with a linear subspace separating answerable from unanswerable questions; probing classifiers can recover this signal, and the authors even explore erasing it [30]. In neural machine translation, Ref. [151] analyzes how specific tokens drive hallucinations via source perturbations, saliency analysis, and Layer-wise Relevance Propagation (LRP).

5.5.2. Neuron Activation and Layer Analysis

Neuron activation and layer analysis (Figure 18) complements internal state probing by shifting from probe classifiers to direct inspection of activations—examining how individual neurons and layers respond to inputs. Early layers tend to encode low-level features, while later layers capture abstract concepts; by tracing activation patterns, researchers identify which components most influence decisions and link meaningful neural activity to behaviors such as honesty, sycophancy, and other traits [112]. Tools include activation map visualizations and attention heatmaps, while layer-wise analyses reveal how feature abstraction progresses through the network, clarifying the internal pathways that can contribute to or mitigate hallucinations [290].

In [290] the researchers investigate how internal activations quantify truthfulness by analyzing local intrinsic dimensions (LID) within these activations. Tracking LID across layers, the authors find increasing LID in early layers, which correlates with hallucinated outputs, and a decrease in later layers. Comparing human-written vs. hallucinated text, human responses show lower LID (more structured), highlighting a divergence between LLM generations and human-like output. Ref. [291] introduces knowledge neurons—units

in feed-forward layers that encode specific factual knowledge. Using integrated gradients for knowledge attribution, the method scores each neuron's contribution to predicting the correct entity in cloze queries. Subsequently, it refines the set by comparing attributions across diverse prompts for the same fact, retaining consistently activated neurons. These identified neurons can be suppressed, amplified, or edited to probe or modify parametric knowledge without full fine-tuning.

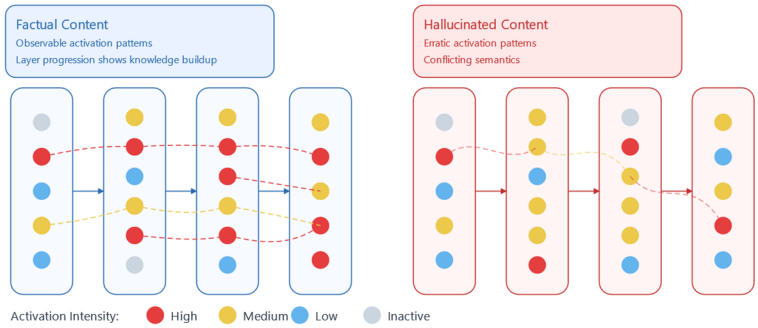

Figure 18. A simplified overview of how neuron activations and progressive layer behaviors reveal the encoding of factual vs. hallucinated content.

Beyond serving as a decoding method, DoLa leverages the tendency for factually correct answers to emerge in later layers, while earlier layers preserve syntactically plausible but incorrect options. It selects a "premature layer" by maximizing Jensen–Shannon divergence from the final layer's distribution, explicitly contrasting deeper, factual signals with earlier, less reliable ones [150]. Related work tracks KL-divergence between intermediate and final layers to quantify how hidden-state distributions evolve with factuality [174]; identifies activation "sharpness" in informative layers and uses cross-layer entropy to separate factual from hallucinated responses [184]; and analyzes attention distributions at the encoder–decoder interface via entropy and cross-attention heads [292]. Logit Lens and Tuned Lens map residual streams to the vocabulary space to visualize how token probabilities change with depth, revealing that correct answers often surface and stabilize later than distractors [293].

Pruning is an effective summarization-time mitigation: by removing redundant weights, it pushes models to lean on the source text, increasing lexical overlap and reducing hallucinations relative to the unpruned baseline [214]. Converging evidence supports a truthfulness signal in model depth: TruthX links intermediate and final layers to factuality [121], while TruthForest exploits activation patterns (notably attention heads and hidden layers) to build orthogonal probes aligned with truthful representations [120]. Complementary diagnostics map where hallucinations arise: layer-wise relevance propagation (LRP) decomposes predictions into layer-level relevance scores to reveal contributing components [151], and neuron-activation analyses surface anomalous patterns indicative of hallucinations or bias [284]. Adaptive Token Fusion (ATF) identifies tokens with high contextual similarity and fuses them into a single, semantically representative token, without losing key meaning [169]. Complementary work [41] perturbs token embeddings and activations to show how changes at these levels alter outputs. Similarly, Ref. [226] traces how token embeddings are transformed across layers to determine where truth-related information is represented in depth.

We conclude this section with a number of papers that apply statistical/mathematical tools to expose latent-space structure linked to hallucinations. PoLLMgraph models temporal state dynamics with Hidden Markov Chains (HMC), binding transitions to a small labeled reference set and successfully detecting hallucinations across diverse scenarios [286]. In parallel, SVD is used to identify principal directions and directly edit neuron activations

at selected layers, steering generation toward desired behavior [180]. Finally, PCA projections of activations reveal that deeper layers separate true vs. false along near-linear axes, which suggests depth-wise organization of truth signals [288].

5.5.3. Attribution-Based Diagnostics

Attribution-based diagnostics trace model outputs to specific inputs or internal steps, assigning importance scores to tokens or features. As diagnostic methods, they often overlap with internal state probing (e.g., gradient attribution and linear probes in [148]) and Neuron Activation and Layer Analysis (e.g., LRP across layers [151]). In this section, we examine those methods from an attributional perspective, and demonstrate how they reveal biases, spurious correlations, or contextually irrelevant dependencies, thus enabling targeted refinements.

Attribution-driven diagnostics link outputs to specific inputs and internal pathways. A Grad-CAM–style method [294] averages input-embedding gradients to score token importance, revealing which inputs shape internal states and downstream hallucination estimates. In MT, ALTI+ [272] defines hallucinations as source-detached translations and computes token-level, layer-wise attributions from each source token to each generated token. Ref. [96] automatically labels unsupported output spans at the token level to flag risk on new inputs. Extending classic attributions, Ref. [295] adapts Gradient $\times$ Input and Input Erasure to produce contrastive explanations, identifying salient tokens that explain why one prediction was preferred over another. Moving from correlation to causation, Ref. [288] shows that adding/subtracting learned "truth" directions in representation space can flip TRUE/FALSE judgments, indicating the directions cause the prediction, not merely correlate with it. Finally, Ref. [296] uses causal mediation analysis and embedding-space projections to derive layer-wise attributions that trace hallucinations to particular attention heads and hidden states, distinguishing early-site from late-site causal contributors. Constrained attention [170] and cross-attention analysis [184] can also function as attribution-based diagnostic tools by examining how LLMs prioritize specific "constraint tokens" within queries. Greater attention to these constraints predicts factual accuracy [170], while probing these internal attention patterns allows for the prediction of factual errors and assessment of factual constraints [184].

Research indicates that statistical regularities and pretraining biases can overshadow relevant conditions, helping explain why certain hallucinations arise [54]. Using PMI, authors pre-identify overshadowed conditions and track probability shifts after selectively removing them—an approach akin to causal probing that attributes errors to specific biases rather than task difficulty [54]. Complementary work ties hallucinations to memorization and corpus-frequency effects; named-entity substitution shows models often use entities as memory indices [36]. Controlled interventions on NLI further suggest many false entailments stem from memorization, not reasoning, functioning as a mechanistic probe of failure modes [35]. Additional "risk factors" include commonsense memorization, relational reasoning, and instruction following [248]. As a black-box diagnostic, patterns in answers to unrelated follow-up questions after a suspected lie reveal signals of deceptive or unstable behavior [117]. Finally, activation-level studies identify pivotal nodes encoding these spurious patterns and show that anomalous sentences correlate with anomalous activations [284].

We conclude this section with Faithful Finetuning (F2), which involves attribution by decomposing QA tasks into Internal Fact Retrieval and Fact-Grounded QA, thereby training the model to explicitly align responses with factual spans [94]. Using entity-based and attention-based heuristics, the authors are able to identify hallucination-prone spans to which they assign higher weights, thus attributing hallucination risk to specific entities.

Ref. [184] broadens the use of entity- and attention-based heuristics to encompass entropy as a metric for diagnosing and detecting factual errors, as ROC-AUC evaluations of entropy as a hallucination indicator resemble attribution-based interpretability methods.

5.6. Agent-Based Orchestration

Agent-based orchestration refers to frameworks that embed single or multiple large language models within multi-step loops, enabling iterative reasoning, tool integration, and dynamic retrieval. These architectures often involve reflexive or self-reflective agents capable of evaluating and refining their own outputs, as well as modular or multi-agent systems in which multiple LLM-based agents interact and collaborate to address complex tasks. Compared to other categories, the current literature on agent-based orchestration is still relatively limited; however, it is expanding rapidly, with growing interest in self-reflective reasoning agents, multi-agent collaboration, and modular orchestration frameworks. The conciseness of this section therefore reflects the novelty of this research direction rather than its potential, as emerging work suggests that agent-based orchestration may become a central paradigm for hallucination mitigation in the near future.

5.6.1. Reflexive/Self-Reflective Agents

Reflexive and self-reflective agents (Figure 19) are systems that both act in real time (reflexivity) and critically evaluate their own behavior, goals, and internal states (self-reflection) [28,45]. While "critical analysis" can appear in CoT-style prompting or in self-reflective loops, we distinguish them by the locus of control: in Structured/Iterative Reasoning Prompting, step-by-step reasoning is embedded in (and triggered by) the user's prompt; in Agent-based Orchestration, the step-by-step process is initiated and managed by the agent, which plans, executes, and reflects over a sequence of actions to reach a goal—either autonomously or in coordination with other agents within a multi-agent system.

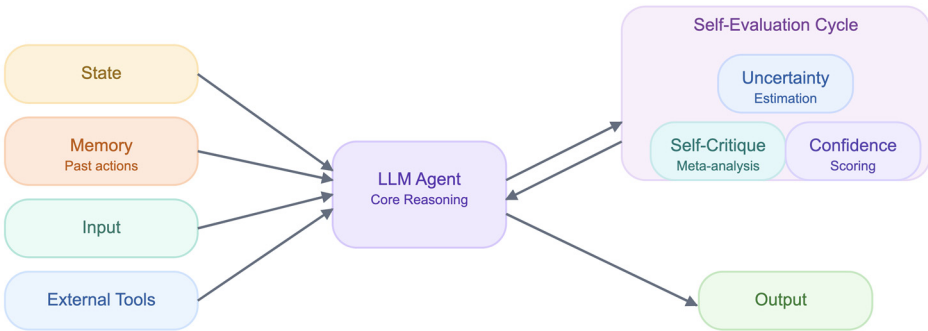

Figure 19. Diagram showing the architecture of a self-reflective agent that utilizes external tools and introspective feedback loops.

Self-reflection appears as an emergent ability mainly at larger scales (≈70B params) [63]; in smaller models it can backfire—second-guessing correct answers and increasing hallucinations [5,63]. Its effectiveness is domain-sensitive: substantial gains in open domains, but only marginal improvements in specialized areas (e.g., finance, science) where self-corrective knowledge is limited [5,63]. Reflexive agents seek rapid, stimulus-driven responses, whereas self-reflective agents trigger reflection via confidence/entropy signals [77,78,155] and are often embedded in iterative reasoning loops that cyclically assess and refine outputs [45]. Within such frameworks, agents analyze decisions, evaluate outcomes, and adapt strategies from prior experience—improving accuracy and contextual relevance over time [28,256]. Ref. [27] studies meta-thinking—self-reflection, self-evaluation, and self-regulation of reasoning—via self-distillation, self-checking, meta-reward formulation, and internal error detection, arguing LLMs can become aware of their own reasoning

chains and improve them [27,45]. CRITIC [256] presents a self-reflective agent that critiques, verifies, and refines its outputs by orchestrating external tools (e.g., Google Search, code interpreters) in a self-corrective loop grounded in external fact-checking. Reflexion [28] generalizes this principle for decision-making, reasoning, and programming: its Self-Reflection model (Msr) generates natural-language critiques of past actions, stores them in memory, and reuses them as feedback to iteratively improve the agent's performance.

Inspired by the concept of self-reflection, Self-Refine models a single LLM playing multiple roles as it generates and reflects on its own output, improving it through structured, multi-round internal dialogue. Unlike frameworks that depend on search engines, Self-Refine models a single LLM that alternates roles—generate → critique → revise—to improve its own output. It is tool-independent and task-agnostic, but depends on the base model's few-shot/instruction-following capability. Empirically, it outperforms one-shot baselines across seven diverse tasks [281]. Graph of Thoughts (GoT) similarly embeds self-evaluation and refinement of intermediate "thoughts." While not a full agentic system, its use of controllers, evaluators, and scoring functions makes it conceptually aligned with reflexive agent architectures [64].

Despite their potential, the development of reflexive and self-reflective agents faces several challenges, including the computational overhead associated with meta-reasoning and the complexity of designing agents capable of balancing reflexivity with reflective deliberation [297]. Nevertheless, advancements in cognitive architectures and meta-learning are progressively addressing these challenges, enabling the development of agents with robust and adaptive intelligence suitable for high-stakes applications [298].

5.6.2. Modular and Multi-Agent Architectures

Beyond self-reflective agents, complex tasks benefit from collaborative intelligence via modular and multi-agent architectures (Figure 20). These frameworks embed LLMs within broader decision pipelines, where specialized agents dynamically invoke external tools (e.g., database query engines, fact-checking APIs), orchestrate multi-step reasoning and retrieval, and maintain dynamic memory of intermediate outcomes to reflect and adapt strategies in real time [109,172,299,300]. Hallucinations are curtailed through cross-verification of agents' outputs and feedback loops before finalization. To address alignment mismatches between human values and training objectives, agents can be trained directly on human preference signals over trajectory segments (e.g., RLHF) or via alternative RL approaches, improving intent alignment and reducing unintended or hallucinated agent behaviors [129,301].

In addition to self-reflective capabilities discussed in the previous section, Ref. [27] introduces multi-agent strategies such as supervisor-agent hierarchies, agent debates, and self-play setups. Specifically, the authors propose a modular decomposition of reasoning tasks where different agents interact through defined roles and communication protocols. These architectures support collaborative meta-reasoning and help overcome the limitations of single-agent feedback loops, especially for high-complexity tasks [27]. Decompose-and-Query (D&Q) [109] operates as an agentic pipeline akin to ReAct, decomposing questions and querying external KBs/tools to harden the LLM against spurious content. A multi-agent debate framework [299] runs independent LLMs that iteratively refine answers via consensus, significantly reducing hallucinations on biographies and MMLU, and outperforming self-reflection approaches [28,281] across six tasks (e.g., arithmetic, biographies, chess). In customer service, Ref. [300] coordinates a Knowledge Retrieval Agent for grounding, a Fuzzy Logic Agent for certainty/truthfulness assessment, and a Validation Agent for external cross-checks, boosting reliability. To stress-test systems, Ref. [139] builds a prompt-based pipeline that induces hallucinations and then assigns two additional detec-

tor/clarifier agents and a fourth KPI evaluator, all orchestrated under the OVON framework for explainability. Finally, RAG-KG-Incremental Learning [189] integrates RAG, KG reasoning, and incremental learning across multiple agents, storing new entities/relations in a KG to expand knowledge continuously.

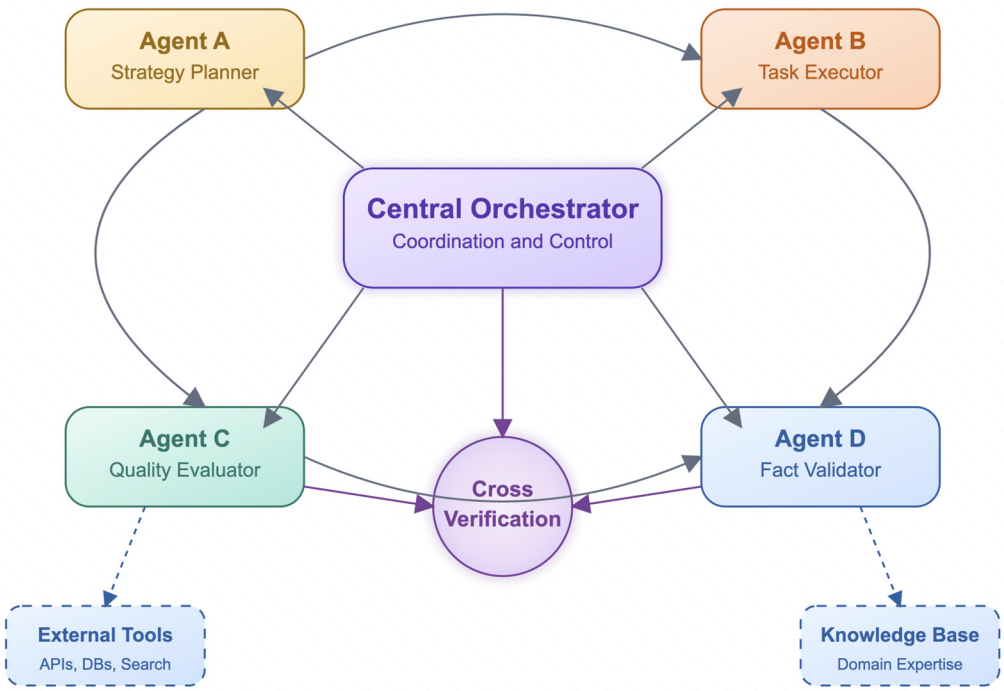

Figure 20. Illustration of multi-agent systems where specialized LLM agents—such as planners, executors, evaluators and validators—collaborate in orchestrated workflows. These architectures may incorporate retrievers, feedback mechanisms, and invocations of external tools.

Conceptually aligned with multi-agent designs, Ref. [251] instantiates the same LLM twice (Examiner/Examinee) with different prompts to simulate modular agents tackling sub-tasks collaboratively. Ref. [22] implements a modular pipeline with a policy agent that orchestrates action sequences per input/task, illustrating the overlap between external verification and orchestration frameworks. Ref. [220] likewise employs agents for detection/mitigation in an orchestration-like setup, though without external tools, dynamic retrieval, or verbose reasoning. RAP (Reasoning via Planning) [233] is explicitly agentic: the LLM acts both as a reasoning agent and as a world model, conducting dynamic exploration with Monte Carlo Tree Search (MCTS) to balance exploration/exploitation while self-guiding along reasoning paths.

Despite successes, current AI agents face notable limits. A narrow accuracy-first focus yields overly complex, costly systems and misleading performance claims [297]. In multi-agent setups, cascading hallucinations can propagate [65], and over-reliance on external tools risks outdated or incomplete evidence [73,114]. Weak holdout sets encourage overfitting, hurting real-world performance, while nonstandard evaluation fuels reproducibility issues and inflated accuracy. Practical deployment must confront scalability, balance autonomy vs. coordination, design efficient communication protocols, and optimize interactions for high-stakes use. Ongoing work—especially in multi-agent RL (MARL)—aims to address these gaps and enable more robust agent architectures [301].

6. Benchmarks for Evaluating Hallucinations

Evaluating hallucinations is intrinsically difficult because they are highly context-dependent. As a result, robust assessment typically combines automated metrics with

task-specific benchmarks: RAG metrics emphasize grounding, semantic entropy captures intrinsic uncertainty, factuality benchmarks [255,301,302] enable fine-grained error typologies, and TruthfulQA probes adversarial robustness. Traditional generation metrics (e.g., METEOR [303], ROUGE [304]), BERTScore [305] and semantic faithfulness evaluations [306] are also reported. Yet no single metric or benchmark fully captures the phenomenon, making human assessment indispensable, especially where context sensitivity and domain expertise are critical [307]. Compounding this, many dialogue benchmarks include hallucinated gold responses (reported as high as 60%) [39], inflating apparent performance and underscoring the need for more reliable datasets. Dialogue summarization presents additional challenges, with models producing plausible but unsupported inferences [308]. Consequently, despite cost and time, human evaluation remains crucial for judging subjective quality and coherence [303,309]. Below, we provide a non-exhaustive list of benchmarks, grouped into Factual Verification, Domain-Specific, and Code Generation categories:

- Factual Verification Benchmarks: These benchmarks focus on assessing the factual accuracy of LLM outputs by comparing them against established ground truth.
    - ANAH is a bilingual dataset for fine-grained hallucination annotation in large language models, providing sentence-level annotations for hallucination type and correction [57]
    - BoolQ: A question answering dataset which focuses on yes/no questions, requiring models to understand the context and before deciding [310].
    - DiaHalu is introduced as the first dialogue-level hallucination evaluation benchmark for LLMs, designed to move beyond purely factual errors by spanning four multi-turn settings—knowledge-grounded, task-oriented, chit-chat, and reasoning [260].
    - FACTOR (Factuality Assessment Corpus for Text and Reasoning) is a benchmark for evaluating LLM factuality with an emphasis on multi-hop reasoning and evidence retrieval [311].
    - FACTSCORE [255] is a fine-grained, atomic-level metric that assesses factual precision in long-form outputs by labeling each claim as supported, unsupported, or unverifiable.
    - FELM (Factuality Evaluation of Large Language Models) is a benchmark dataset for testing factuality evaluators on long-form LLM outputs across five domains—world knowledge, science/tech, math, reasoning, and writing/recommendation—by measuring their ability to detect factual errors [312].
    - FEVER (Fact Extraction and Verification): FEVER is a 185,445-claim dataset that serves as a challenging benchmark—requiring multi-hop reasoning and evidence retrieval—to test models' ability to gather relevant evidence and determine claim veracity [313].
    - FEWL (Factuality Evaluation Without Labels): FEWL is a methodology for measuring and reducing hallucinations in large language models without relying on gold-standard answers [243].
    - The FRANK benchmark for abstractive summarization provides fine-grained error annotations on summaries from nine systems, enabling rigorous evaluation and comparison of factuality metrics [302].
    - HADES (HAllucination DEtection dataset) is a reference-free hallucination-detection dataset for QA, built by perturbing Wikipedia text and human-annotating via a model-in-the-loop process, enabling detection of hallucinations without ground-truth references [85].

- HalluEditBench is a dataset comprising verified hallucinations across multiple domains and topics and measures editing performance across Efficacy, Generalization, Portability, Locality, and Robustness [314].
- HalluLens is a hallucination-focused benchmark that covers intrinsic and extrinsic tasks, dynamically generates test sets to curb data leakage, and aims for task-aligned detection by treating hallucinations as inconsistency with training/user input rather than absolute truth [315].
- HALOGEN (Hallucinations of Generative Models) is a multi-domain hallucination benchmark that evaluates LLMs on hallucination frequency, refusal behavior, and utility, shedding light on error types and their likely pretraining-data sources [316].
- HaluEval: HaluEval is a large-scale benchmark designed to evaluate the hallucination tendencies of large language models (LLMs). It measures how well LLMs can generate factually accurate content and identify information that is hallucinated or incorrect [224].
- HaluEval 2.0: HaluEval 2.0 is an enhanced version of the original HaluEval benchmark, containing 8770 questions from diverse domains offering wider coverage and more rigorous evaluation metrics for assessing factuality hallucinations [5].
- HaluEval-Wild: HaluEval-Wild is a benchmark designed to evaluate hallucinations within dynamic, real-world user interactions as opposed to other benchmarks that focus on controlled NLP tasks like question answering or summarization [107].
- HDMBench is a benchmark designed for hallucination detection across diverse knowledge-intensive tasks [111]. It includes span-level and sentence-level annotations, covering hallucinations grounded in both context and common knowledge.
- Head-to-Tail: Head-to-Tail delves into the nuances of factual recall by categorizing information based on popularity [203]. It consists of 18,000 question-answer pairs and segments knowledge according to popularity.
- HotpotQA: HotpotQA evaluates multi-hop reasoning and information retrieval capabilities, requiring models to synthesize information from multiple documents to answer complex questions [144] which intersect with broader concerns about faithfulness and hallucinations addressed in [206].
- NQ (Natural Questions): NQ is a large-scale dataset of real questions asked by users on Google, paired with corresponding long-form answers from Wikipedia [317]. It tests the ability to retrieve and understand information from a large corpus.
- RAGTruth is a dataset tailored for analyzing word-level hallucinations within standard RAG frameworks for LLM applications. It comprises nearly 18,000 naturally generated responses from various LLMs that can also be used to benchmark hallucination frequencies [114].
- SelfCheckGPT is a zero-resource, black-box benchmark for hallucination detection. It assesses model consistency by sampling multiple responses and measuring their similarity, without needing external databases or model internals [267].
- TriviaQA is a question-answering dataset that contains over 650,000 question-answer-evidence triplets that were created by combining trivia questions from various web sources [318].
- TruthfulQA: TruthfulQA is a benchmark designed to assess the capability of LLMs in distinguishing between truthful and false statements, particularly those crafted to be adversarial or misleading [53].

  - The UHGEval benchmark offers a large-scale, Chinese-language dataset for evaluating hallucinations under unconstrained generation settings. UHGEval captures naturally occurring hallucinations from five LLMs and applies a rigorous annotation pipeline, making it a more realistic and fine-grained resource for factuality evaluation [319].
- Domain-Specific Benchmarks: These benchmarks target specific domains, testing the model's knowledge and reasoning abilities within those areas.
  - PubMedQA: This benchmark focuses on medical question answering, evaluating the accuracy and reliability of LLMs in the medical domain [240].
  - SciBench: This benchmark verifies scientific reasoning and claim consistency, assessing the ability of LLMs to understand and apply scientific principles [320].
  - LegalBench: This benchmark examines legal reasoning and interpretation, evaluating the performance of LLMs on legal tasks [10].
- Code Generation Benchmarks (e.g., HumanEval, Codex): These benchmarks assess the ability of LLMs to generate correct and functional code, which requires both factual accuracy and logical reasoning [230].

These benchmarks, summarized in Appendix B, critically examine hallucination phenomena across multiple dimensions: factual verification, contextual coherence, domain-specific reliability, and cross-modal consistency. Arguably, the most sophisticated benchmarking strategies integrate interdisciplinary perspectives, drawing from epistemology, cognitive science, and computational linguistics in order to create assessment frameworks that capture the multifaceted nature of hallucinations.

Benchmark Selection Guidance

Effective benchmark selection aligns with a system's grounding mechanisms and anticipated failure modes. Retrieval-augmented architectures benefit from document-grounded QA tasks featuring gold evidence, where performance evaluation encompasses claim-level entailment checking, citation accuracy, and abstention policies for unsupported queries. Source-conditioned summarization reveals more through datasets that directly measure faithfulness rather than relying solely on surface-level overlap metrics.

Instruction-following and open-domain systems expose their limitations when truthfulness assessments combine with calibration measures, illuminating the critical distinction between uncertainty and confident misinformation. Multi-hop reasoning tasks become most diagnostic when they surface intermediate reasoning chains, enabling evaluation of both logical progression and final outputs. Domain-specific applications in healthcare or legal contexts demand corpora with authoritative, timestamped references, where precision metrics at controlled abstention rates intersect with temporal sensitivity analysis. Data-to-text and table-to-text generation tasks achieve maximum transparency when claims maintain traceable connections to their originating structured fields, facilitating granular correctness assessment.

When choosing among alternatives, users may want to consider: (i) grounding fit (evidence modality matches the target system); (ii) error granularity (supports claim-level scoring, not only document-level overlap); (iii) risk class (high-stakes tasks should mandate abstention and report verifier precision/recall); (iv) recency/drift (temporal splits or freshness checks for volatile facts); (v) factor disentanglement (for RAG, separate retrieval recall from generation faithfulness); and (vi) practicality (dataset size that permits targeted human spot-checks of automatic metrics).

7. Practical Implications

While benchmarks specify what to measure, high-stakes deployment also demands operational guardrails. Drawing on the evidence reviewed, we propose practice-oriented heuristics for deploying hallucination-mitigation in domains such as healthcare, medicine, and defense. These are defaults, not mandates: they must be adapted to each application's risk profile, and validated through local evaluation and human-in-the-loop oversight. The guidance distills recurring patterns and explicitly maps benchmark dimensions to concrete design and engineering decisions [53,85,107,224,255,312,313].

- Make evidence the default: Prefer retrieval-augmented and evidence-linked generation, preferably from well curated knowledge bases; surface source spans for non-trivial claims. Favor designs that bind decoding to retrieved context and revision passes [154,176,193,199,274].
- Calibrate—and abstain when needed: Expose uncertainty (entropy/semantic entropy, PMI-based signals, internal-state metrics) and combine with self-consistency; enforce a strict abstain/deferral policy below thresholds [21,25,26,54,132,275].
- Constrain decoding with verification in the loop: Use conservative decoding and inference-time checks; rerank with faithfulness signals before release [100,119,150,166,184].
- Evaluate with task-relevant metrics: Report faithfulness and grounding alongside utility; do not rely on surface metrics alone [138,154,176,193,199,274,302,321–329].
- Add post hoc fact-checking by default: Run external verification on claims and record provenance; separate self-verification from evidence-based checks to avoid false reassurance [61,188,207,209,230,256,312,313].
- Keep humans in the loop with role clarity: Use Self-Reflection and external fact-checking pipelines to route low-confidence or conflicting outputs to a designated reviewer; require dual sign-off for irreversible actions when sources disagree.
- Log everything for audits: Persist prompts, retrieved docs, model/version, verifier scores, and decision outcomes to enable incident analysis and rollback with pipelines such as [114,190,274].

7.1. Operationalization in High-Stakes Domains

In high-stakes fields, consider implementing conservative, auditable pipelines.

- Healthcare Applications: For healthcare, a retrieve → generate → verify → abstain/revise workflow seems appropriate. The retrieval step could be restricted to reliable, up-to-date sources, with citations provided at the span level. It may be wise to constrain the model's generation to only the retrieved information. Low confidence might trigger a human review, and final outputs should explicitly note any uncertainty. End-to-end auditability can be maintained by logging prompts, model versions, retrieved passages, and final decisions. There are trade-offs to consider, such as balancing accuracy and safety against latency and computational cost.
- Legal Applications: For legal contexts, a workflow of scoped retrieve → structured reasoning → cite-check → redline might be preferred. Retrieval could be limited by jurisdiction and authority. Prompts might enforce a structured analysis, and a secondary checker could validate quotes against authoritative texts. Provenance and rationale should be logged to support audits. Key tensions here could include balancing citation precision against coverage and rigorous verification against speed.
- General Deployment Practices: Across these settings, you may want to locally calibrate retrieval depth, verification thresholds, and abstention policies. It could be beneficial to pilot evaluations that report not only accuracy but also a triplet of outcomes: factuality, robustness to noisy inputs, and latency. In practice, modular or agentic designs that separate the generate, verify, and refine stages can provide better control and

traceability, though this comes with added complexity. These costs should be explicitly documented in a safety case.

7.2. Does RAG Help or Harm?

In Section 5.2.3, where we analyzed RAG in relation to hallucinations, we noted that its use can, in some settings, be counterproductive. For researchers and practitioners considering RAG, we offer a concise, practice-oriented guide. Empirically, RAG tends to reduce hallucinations when three preconditions hold: (i) retrieval is high-precision and up-to-date (relevant, low-noise sources with adequate recall), (ii) the generator is tightly conditioned on retrieved spans (via focused chunking/reranking that minimizes context dilution), and (iii) claim-level verification is enforced with an abstention or deferral policy when support is absent. Under these conditions, unsupported assertions are either avoided or revised against evidence, and grounding cues effectively constrain decoding. Conversely, hallucination rates increase when any precondition fails—e.g., imprecise or stale retrieval steers generations toward incorrect facts; indiscriminate or excessive context weakens conditioning; complex or multi-hop tasks amplify spurious correlations introduced by retrieval; and pipelines lacking claim–evidence checks allow unsupported spans to pass unchecked. The benefit of RAG is therefore contingent not on "using RAG" per se, but on how retrieval is timed, filtered, and integrated.

Operationally, explicit locally validated criteria outperform universal thresholds. Target verification precision drives acceptance/abstention tuning, ensuring claims emerge only when supporting spans demonstrate clear entailment with retrieved evidence. Context growth requires boundaries through limited values and controlled chunk lengths, while reranking elevates high-yield passages and prevents dilution. Risk signals should be used to gate retrieval processes rather than allowing unconditional access. A triplet of outcomes demands monitoring on local development sets: factuality/faithfulness, robustness against noisy or outdated inputs, and compute performance.

7.3. Computational Trade-Offs

In addition to the high-level practices mentioned previously, practitioners often need concrete guidance on the computational costs of alternative mitigation strategies. Accordingly, we have compiled Table 1 as a practical guideline for users and researchers, comparing strategies by indicative latency and memory proxies (e.g., additional LM calls, context growth, and iteration depth) while abstracting away hardware/model specifics:

Table 1. Indicative computational footprint of hallucination-mitigation strategies.

Strategy	Extra LM Calls per Query	Context Growth	Other Modules	Latency Impact	Memory Impact	Notes
Prompt Engineering	+0	±0	—	Low	Low	Sensitive to prompt design; cheapest mitigation.
Decoding constraints/contrastive decoding	+0	±0	per-token ops	Low–Med	Low	Per-token overhead (≈10–40%) depending on constraint strength.
RAG (BM25/embedding + reranker + generator)	+0–1 (reranker)	High (k × chunk_len)	retriever, index I/O	Med–High	Med	Cost scales with k (docs) and chunk size; reranker adds one pass.

Table 1. *Cont.*

Strategy	Extra LM Calls per Query	Context Growth	Other Modules	Latency Impact	Memory Impact	Notes
RAG + post-gen verifier (claim checker, NLI)	+1 (verifier)	High	retriever + verifier	High	Med–High	Better precision; ~1 extra LM pass for verification.
Self-verification/critic (same/smaller LM)	+1 (critic)	±0	—	Med–High	Low–Med	One additional pass; loops increase cost linearly.
Agentic pipelines	+m (m stages)	High	tool calls/IO	High–Very High	Med–High	Cost ≈ m × single-pass + retrieval/verification overheads.

8. Extended Discussion

Our taxonomy focuses on mechanisms rather than tasks. Across the six categories of our taxonomy, a consistent pattern emerges: Training-and-learning methods buy reliability upfront: they are expensive offline but light at inference time, and they tend to travel well across tasks when supervision encodes uncertainty rather than certainty. Architectural modifications give direct control—steering attention, shaping decoding, or adding retrieval—yet they live or die by the timing and quality of what is injected into the model. Prompt and input optimization is the most economical knob to turn and often delivers surprising gains, but it is sensitive to phrasing and can drift over long conversations. Post-generation checks, from self-verification to external fact-checking, deliver the cleanest precision and clear audit trails; the price is extra passes and slower responses. Interpretability and diagnostics do not usually change outputs on their own, but they tell us where truth lives in the network and where errors start, enabling safer edits later. Agent-based orchestration integrates many of these ideas—decomposing tasks, cross-checking answers, and calling tools—but introduces the most complexity and the clearest need for process-level evaluation. We have summarized these findings in the Appendix B (Figure A1).

The computational picture mirrors these narratives (Table 1). In practical terms, prompt engineering and light decoding constraints are the fastest options. Retrieval-augmented generation improves grounding but carries the cost of fetching and feeding more context. Adding a verifier after generation sharpens accuracy yet typically adds another model pass. Self-verification loops scale with the number of revisions. Full agentic pipelines, with multiple stages and tool calls, trade speed for traceability and control. These are not hardware-specific measurements; they are simple waypoints for planning systems that must balance responsiveness with rigor.

Two trade-offs recur regardless of method: precision versus speed, and control versus complexity. Systems that cite sources and check claims are slower but easier to audit. Systems that decompose and debate reach better answers but require stronger engineering discipline to prevent cascade errors. For that reason, we encourage reporting results as a three-part outcome—faithfulness/grounding, robustness to noisy or outdated inputs, and latency/compute—so improvements in truthfulness are not purchased at an invisible operational cost.

9. Challenges

The mitigation of hallucinations in large language models (LLMs) remains a multifaceted challenge situated at the intersection of deep learning, cognitive science, and epistemology. While significant progress has been made, several open issues continue to limit the robustness, generalizability, and ethical deployment of current approaches. Drawing on the synthesis of more than 300 studies reviewed in this work, we consolidate these findings into the most prominent challenges, which are outlined in the issue-based discussion below.

- Lack of Standardized Benchmarks and Metrics: The evaluation of hallucination mitigation is fragmented, using ad hoc datasets and inconsistent metrics. This makes it difficult to compare different methods. Creating shared benchmarks that include various hallucination types and languages is necessary for establishing reliable baselines and systematic evaluation.
- Interpretability and Attribution Difficulties: It is challenging to understand how transformer-based models create hallucinations due to their complexity. This makes it hard to pinpoint the cause of errors, especially in multi-method systems. Improved interpretability tools and causal analysis are crucial for building trust in these applications.
- Robustness Under Distribution Shifts and Adversarial Inputs: Large Language Models (LLMs) are often vulnerable to hallucinations when they encounter data or prompts that differ from their training data. Ensuring the models are resilient under these conditions requires improved uncertainty estimation and new training methods.
- Computational Trade-offs and Latency Constraints: Many mitigation strategies, such as retrieval-augmented generation, add significant computational overhead, creating a conflict between accuracy and speed. Future research must focus on efficient mitigation techniques that are both reliable and practical for real-world use.
- Knowledge Limitations and Updating: Hallucinations often happen because models use outdated or incomplete training data. While retrieval-based methods help, they can still be affected by errors in external sources. More robust strategies for continuously updating knowledge and noise-aware retrieval are needed.
- Ethical and Epistemological Concerns: Hallucination mitigation is also an ethical issue. Researchers must navigate the line between factual reliability and creative generation, especially in sensitive fields like healthcare and law. This requires both technical safeguards and governance frameworks.

These challenges show that solving hallucination mitigation is not about simple fixes, but about tackling a set of interconnected issues related to data, architecture, evaluation, and ethics. Addressing them requires coordinated progress, not piecemeal solutions. This review offers a dual perspective: a diagnostic lens on the limitations of current approaches and a roadmap for future research. As the field grows, particularly in areas like multi-agent orchestration, hybrid pipelines, and knowledge-grounded fine-tuning, this framework can help guide the development of more reliable and transparent LLMs for important applications.

10. Conclusions

Our review reveals that large language model hallucinations are caused by interacting mechanisms rather than a single failure. The originality of our work lies in a method-oriented taxonomy that organizes mitigation strategies based on the operational levers they activate across the model lifecycle—including training, architecture, decoding, retrieval, quality control, and diagnostics—instead of by task or application. This framework goes beyond simple classification; it exposes shared interfaces between methods, clarifies hybrid techniques, and converts a previously scattered literature into a coherent design space. Ultimately, by mapping these mechanisms to evaluation targets and deployment guardrails, the taxonomy acts as an analytical scaffold that enables cumulative and testable progress, making it a dynamic tool instead of a static survey.

Based on this analytical framework, we propose a nine-point research agenda focused on reducing large language model hallucinations and improving trustworthiness: Standardized evaluation must first be developed, moving toward uncertainty-aware metrics that explicitly link factuality with calibration, requiring the model to either abstain or show its sources for every claim. A critical step is the creation of mechanisms that can identify

high-confidence hallucinations so that those errors are converted into a clear refusal or a deferral to a more reliable source. To support this, the model's knowledge flow must be better controlled by formalizing retrieval and internal controls must be advanced with strong safety constraints and robustness checks. Furthermore, alignment learning must be evolved beyond simple human feedback (RLHF) so that verifiable evidence signals are used to directly reward grounded, truthful reasoning. Other key areas require the development of sophisticated agentic systems capable of preventing cascade failures, with decision-making processes being logged. Also, domain-specific protocols—like requiring span-level citation in legal or medical texts—must be codified, complete with clear safety cases. Finally, all these advancements must be made practical by integrating data governance (handling bias, privacy, and licensing) into mitigation pipelines and by prioritizing operational efficiency so that these safety measures do not introduce unacceptable latency or energy costs.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/ai6100260/s1, PRISMA 2020 Checklist [330].

Author Contributions: All authors have contributed to the review presented. Conceptualization, writing—original draft preparation, writing—review and editing, visualization, I.K.; supervision, E.A., K.D. and C.B. All authors have read and agreed to the published version of the manuscript.

Funding: This review article has not received external funding.

Data Availability Statement: Data are contained within the review article.

Conflicts of Interest: The authors declare no conflicts of interest.

Glossary

A	
ActLCD	Active Layer-Contrastive Decoding—Decoding that contrasts intermediate layers to steer token selection toward more factual continuations.
Activation Decoding	Constrained decoding that adjusts next-token probabilities using activation/uncertainty signals to suppress hallucinations.
AFHN	Adversarial Feature Hallucination Networks—Adversarial training to produce features and examples that stress models and reduce hallucinations.
AggTruth	Attention aggregation across heads/layers to flag unsupported spans and improve factual consistency checks.
ALIGNed-LLM	Aligns external knowledge (e.g., KG/entity embeddings) with the model's representation space to ground generations.
ALTI+	Attribution method that quantifies how much each input token contributes to generated tokens for interpretability/factuality analysis.
ANAH	Bilingual hallucination dataset with sentence-level annotations and suggested corrections.
ATF	Adaptive Token Fusion—Merges redundant/similar tokens early to retain meaning while reducing noise and hallucination risk.
AutoHall	Automatic pipeline to synthesize, detect, and evaluate hallucinations for training and benchmarking.
AutoRAG-LoRA	Lightweight LoRA adaptation to better couple retrieval and generation in RAG systems.
B	
BERTScore	Semantic similarity metric using contextual embeddings to evaluate generated text.
BLEU	N-gram overlap metric; useful for surface similarity but not a direct measure of factuality.
BoolQ	Yes/no question-answering dataset often used in factuality experiments.

C	
CCL-XCoT	Cross-lingual transfer of Chain of Thought traces to improve reasoning and reduce hallucinations across languages.
CD	Contrastive Decoding—Penalizes tokens favored by a weaker/contrast model to filter implausible continuations.
CoK	Chain of Knowledge—Grounds reasoning by explicitly incorporating external knowledge into intermediate steps.
COMET-QE	Reference-free MT quality estimation used as a proxy signal for consistency.
Confident Decoding	Incorporates uncertainty estimates into beam/nucleus procedures to favor low-uncertainty continuations.
CoNLI	Chain of NLI—Cascaded entailment checks over partial outputs to prune unsupported content.
CoT	Chain of Thought—Prompting that elicits step-by-step reasoning before the final answer.
CoVe	Chain of Verification—Contextualized word embeddings pre-trained on translation, used for stronger semantic representations.
CPM	Conditional Entropy Mechanism—Uses token-level entropy to detect and avoid uncertain/hallucination-prone outputs.
CPMI	Conditional Pointwise Mutual Information—Decoding re-scoring that rewards tokens better supported by the source/context.
CPO	Contrastive Preference Optimization—Preference optimization that uses contrastive signals to align outputs with faithful behavior.
CRAG	Corrective Retrieval-Augmented Generation—Adds corrective/revision steps atop RAG to fix unsupported claims.
CRITIC	A verify-and-edit framework where a "critic" process checks claims against evidence and proposes fixes.
Critic-driven Decoding	Decoding guided by a trained critic/verifier that down-weights unsupported next tokens.
D	
D&Q	Decompose-and-Query—Decomposes a question into sub-questions and retrieves evidence for each before answering.
DeCoRe	Decoding by Contrasting Retrieval Heads—Contrasts retrieval-conditioned signals to suppress ungrounded tokens.
Dehallucinator	Detect-then-rewrite approach that edits hallucinated spans into grounded alternatives.
Delta	Compares outputs under masked vs. full context to detect and penalize hallucination-prone continuations.
DiaHalu	Dialogue-level hallucination benchmark covering multiple multi-turn domains.
DoLa	Decoding by Contrasting Layers—Uses differences between early vs. late layer logits to promote factual signals.
DPO	Direct Preference Optimization—RL-free preference tuning that directly optimizes for chosen responses.
DRAD	Decoding with Retrieval-Augmented Drafts—Uses retrieved drafts/evidence to guide decoding away from unsupported text.
DreamCatcher	Detects and corrects hallucinations by cross-checking outputs against external evidence/tools.
DrHall	Lightweight, fast hallucination detection targeted at real-time scenarios.
E	
EigenScore	Uncertainty/factuality signal derived from the spectrum of hidden-state representations.
EntailR	Entailment-based verifier used to check whether generated claims follow from retrieved evidence.

EVER	Evidence-based verification/rectification that validates claims and proposes fixes during/after generation.
F	
F2	Faithful Finetuning—Direct finetuning objective to increase faithfulness of generations.
FacTool	Tool-augmented factuality checking that extracts claims and verifies them against sources.
FactPEGASUS	Summarization variant emphasizing factual consistency with the source document.
FactRAG	RAG design focused on retrieving and citing evidence that supports each claim.
FACTOR	Benchmark emphasizing multi-hop factuality and evidence aggregation.
FAVA	Corrupt-and-denoise training pipeline to teach models to correct fabricated content.
FELM	Benchmark for evaluating factuality evaluators on long-form outputs.
FEVER	Large-scale fact verification dataset (Supported/Refuted/Not Enough Info).
FG-PRM	Fine-Grained Process Reward Model—Process-level reward modeling for stepwise supervision of reasoning.
FRANK	Fine-grained factual error taxonomy and benchmark for summarization.
FreshLLMs	Uses live retrieval/search refresh to reduce outdated or stale knowledge.
FactScore	Atomic, claim-level factuality scoring/benchmark for long-form text.
G	
GAN	Generative Adversarial Network—Adversarial training framework used to stress and correct model behaviors.
GAT	Graph Attention Network—Graph neural network with attention; used to propagate grounded evidence.
GNN	Graph Neural Network—Neural architectures over graphs for structured reasoning/grounding.
GoT	Graph-of-Thoughts—Represents reasoning as a graph of states/operations to explore multiple paths.
Grad-CAM	Gradient-based localization on intermediate features for interpretability of decisions.
Gradient $\times$ Input	Simple attribution method multiplying gradients by inputs to estimate token importance.
Graph-RAG	RAG that leverages knowledge graphs/graph structure for retrieval and grounding.
G-Retriever	Graph-aware retriever designed to recall evidence that reduces hallucinations.
H	
HADES	Reference-free hallucination detection dataset for QA.
HALO	Estimation and reduction framework for hallucinations in open source LLMs.
HALOGEN	Structure-aware reasoning/verification pipeline to reduce unsupported claims.
HalluciNot	Retrieval-assisted span verification to detect and mitigate hallucinations.
HaluBench	Benchmark suite for evaluating hallucinations across tasks or RAG settings.
HaluEval	Large-scale hallucination evaluation benchmark.
HaluEval-Wild	"In-the-wild" hallucination evaluation using web-scale references.
HaluSearch	Retrieval-in-the-loop detection/mitigation pipeline that searches evidence while generating.
HAR	Hallucination Augmented Recitations—Produces recitations/snippets that anchor generation to evidence.
HDM-2	Hallucination Detection Method 2—Modular multi-detector system targeting specific hallucination types.

HERMAN	Checks entities/quantities in outputs against source to avoid numerical/entity errors.
HILL	Human-factors-oriented hallucination identification framework/benchmark.
HIPO	Hard-sample-aware iterative preference optimization to improve robustness.
HMC	Hidden Markov Chains—Sequential state models used to analyze latent dynamics associated with hallucinations.
HSP	Hierarchical Semantic Piece—Hierarchical text segmentation/representation to stabilize retrieval and grounding.
HybridRAG	Combines multiple retrieval sources/strategies (e.g., dense + sparse + KG) for stronger grounding.
HumanEval	Code generation benchmark often used in hallucination-sensitive program synthesis.
HVM	Hypothesis Verification Model—Classifier/verifier that filters candidates by textual entailment with evidence.
I	
ICD	Induce-then-Contrast Decoding—Induces errors with a weaker model and contrasts to discourage hallucinated tokens.
INSIDE	Internal-state-based uncertainty estimation with interventions to reduce overconfidence.
Input Erasure	Attribution by removing/ablating input spans to see their effect on outputs.
InterrogateLLM	Detects hallucinations via inconsistency across multiple answers/contexts.
Iter-AHMCL	Iterative decoding with hallucination-aware contrastive learning to refine outputs.
ITI	Inference-Time Intervention—Nudges specific heads/activations along truth-aligned directions during decoding.
J	
Joint Entity and Summary Generation	Summarization that jointly predicts entities and the abstract to reduce unsupported content.
K	
KB	Knowledge Base—External repository of facts used for grounding/verification.
KCA	Knowledge-Consistent Alignment—Aligns model outputs with retrieved knowledge via structured prompting/objectives.
KG	Knowledge Graph—Graph-structured facts used for retrieval, verification, and attribution.
KGR	Knowledge Graph Retrofitting—Injects/retrofits KG-verified facts into outputs or intermediate representations.
KL-divergence	Divergence measure used in calibration/regularization and to compare layer distributions.
Knowledge Overshadowing	When parametric priors dominate over context, causing the model to ignore given evidence.
L	
LaBSE	Multilingual sentence encoder used for cross-lingual matching/verification.
LASER	Language-agnostic sentence embeddings for multilingual retrieval/entailment.
LAT	Linear Artificial Tomography—Linear probes/edits to reveal and steer latent concept directions.
LayerSkip	Self-speculative decoding with early exits/verification by later layers.
LID	Local Intrinsic Dimension—Dimensionality measure of hidden states linked to uncertainty/truthfulness.
LinkQ	Forces explicit knowledge-graph queries to ground answers.

LLM Factoscope	Probing/visualization of hidden-state clusters to distinguish factual vs. fabricated content.
LLM-AUGMENTER	Orchestrates retrieval/tools around an LLM to improve grounding and reduce errors.
Logit Lens	Projects intermediate residual streams to the vocabulary space to inspect token preferences.
Lookback Lens	Attention-only method that checks whether outputs attend to relevant context.
LoRA	Low-rank adapters for efficient finetuning, commonly used in factuality/hallucination pipelines.
LQC	Lightweight Query Checkpoint—Predicts when a query needs verification or retrieval before answering.
LRP	Layer-wise Relevance Propagation—Decomposes predictions to attribute token-level contributions.
M	
MARL	Multi-Agent Reinforcement Learning—Multiple agents coordinate/critique each other to improve reliability.
MC	Monte Carlo—Stochastic sampling used for uncertainty estimation and search.
MCTS	Monte Carlo Tree Search—Guided tree exploration used in deliberate, plan-and-verify reasoning.
METEOR	MT metric leveraging synonymy/stemming; not a direct factuality measure.
mFACT	Decoding-integrated factuality signal to prune low-faithfulness candidates.
MixCL	Mixed contrastive learning (with hard negatives) to reduce dialog hallucinations.
MoCo	Momentum contrast representation learning used to build stronger encoders.
MoE	Mixture-of-Experts—Sparse expert routing to localize knowledge and reduce interference.
N	
NEER	Neural evidence-based evaluation/repair methods that use entailment or retrieved evidence to improve outputs.
Neural Path Hunter	Analyzes reasoning paths/graphs to locate error-prone segments for correction.
Neural-retrieval-in-the-loop	Integrates a trainable retriever during inference to stabilize grounding.
NL-ITI	Nonlinear version of ITI with richer probes and multi-token interventions.
NLU	Natural Language Understanding—Models/components (e.g., NLI, QA) used as verifiers or critics.
Nucleus Sampling	Top-p decoding that samples from the smallest set whose cumulative probability exceeds p.
O	
OVON	Open-Vocabulary Object Navigation; task setting where language directs navigation to open-set objects, used in agent/LLM evaluations.
P	
PCA	Principal Component Analysis—Projects activations to principal subspaces to analyze truth/lie separability.
PGFES	Psychology-guided two-stage editing and sampling along "truthfulness" directions in latent space.
Persona drift	When a model's stated persona/stance shifts across sessions or contexts.
PoLLMgraph	Probabilistic/graph model over latent states to track hallucination dynamics.
PMI	Pointwise Mutual Information—Signal for overshadowing/low-confidence conditions during decoding.
Principle Engraving	Representation-editing to imprint desired principles into activations.

Principle-Driven Self-Alignment	Self-alignment method that derives rules/principles and tunes behavior accordingly.
ProbTree	Probabilistic Tree-of-Thought—ToT reasoning with probabilistic selection/evaluation of branches.
PURR	Trains on corrupted vs. corrected claims to produce a compact, factuality-aware model.
TOPICPREFIX	Prompt/prefix-tuning scheme to stabilize topic adherence and reduce drift.
Q	
Q^2	Factual consistency measure comparing outputs to retrieved references.
R	
R-Tuning	Tuning models to abstain or say "I don't know" when unsure.
RAG	Retrieval-Augmented Generation—Augments generation with document retrieval for grounding.
RAG-KG-IL	RAG integrated with knowledge-graph and incremental-learning components.
RAG-Turn	Turn-aware retrieval for multi-turn tasks.
RAGTruth	Human-annotated data for evaluating/teaching RAG factuality.
RAP	Reasoning viA Planning—Planning-style reasoning that structures problem solving before answering.
RARR	Retrieve-and-Revise pipeline that edits outputs to add citations and fix unsupported claims.
RBG	Read-Before-Generate—Reads/retrieves first, then conditions generation on the evidence.
REPLUG	Prepends retrieved text and averages probabilities across retrieval passes to ground decoding.
RepE	Representation Engineering—Editing/steering latent directions to improve honesty/faithfulness.
RefChecker	Reference-based fine-grained hallucination checker and diagnostic benchmark.
Reflexion	Self-critique loop where the model reflects on errors and retries.
RID	Retrieval-In-Decoder—Retrieval integrated directly into the decoder loop.
RHO	Reranks candidates by factual consistency with retrieved knowledge or graph evidence.
RHD	Real-time Hallucination Detection—Online detection and optional self-correction during generation.
RLCD	Reinforcement Learning with Contrastive Decoding—RL variant that pairs contrastive objectives with decoding.
RLHF	Reinforcement Learning from Human Feedback—Uses human preference signals to align model behavior.
RLAIF	Reinforcement Learning from AI Feedback—Uses AI-generated preference signals to scale alignment.
RLKF	Reinforcement-Learning-based Knowledge Filtering that favors context-grounded generation.
ROUGE	Overlap-based summarization metric (e.g., ROUGE-L).
RaLFiT	Reinforcement-learning-style fine-tuning aimed at improving truthfulness/factuality.
S	
SC2	Structured Comparative Reasoning—Compares structured alternatives and selects the most consistent one.
SCOTT	Self-Consistent Chain-of-Thought Distillation—Samples multiple CoTs and distills the consistent answer.
SCD	Self-Contrastive Decoding—Penalizes over-represented priors to counter knowledge overshadowing.
SEA	Spectral Editing of Activations—Projects activations along truth-aligned directions while suppressing misleading ones.

SEAL	Selective Abstention Learning—Teaches models to abstain (e.g., emit a reject token) when uncertain.
SEBRAG	Structured Evidence-Based RAG—RAG variant that structures evidence and grounding steps.
SEK	Evidence selection/structuring module used to verify or revise outputs.
SEPs	Semantic Entropy Probes—Fast probes that estimate uncertainty from hidden states.
Self-Checker	Pipeline that extracts and verifies claims using tools or retrieval.
Self-Checks	Generic self-verification passes (consistency checks, regeneration, or critique).
Self-Consistency	Samples multiple reasoning paths and selects the majority-consistent result.
Self-Familiarity	Calibrates outputs based on what the model "knows it knows" vs. uncertain areas.
Self-Refine	Iterative refine-and-feedback loop where the model improves its own draft.
Self-Reflection	The model reflects on its reasoning and revises responses accordingly.
SELF-RAG	Self-reflective RAG where a critic guides retrieval and edits drafts.
SelfCheckGPT	Consistency-based hallucination detector using multiple sampled outputs.
SH2	Self-Highlighted Hesitation—Injects hesitation/abstention mechanisms at uncertain steps.
SimCLR	Contrastive representation learning framework used to build stronger encoders.
SimCTG	Contrastive text generation that constrains decoding to avoid degenerate outputs.
Socratic Prompting	Uses guided questions to elicit intermediate reasoning and evidence.
SVD	Singular Value Decomposition—Matrix factorization used to analyze or edit latent directions.
T	
ToT	Tree-of-Thought—Branch-and-evaluate reasoning over a tree of intermediate states.
TOPICPREFIX	Prompt/prefix-tuning that encodes topics to stabilize context adherence.
TrueTeacher	Teacher-style training that builds a factual evaluator and uses it to guide student outputs.
Truth Forest	Learns orthogonal "truth" representations and intervenes along those directions.
TruthfulQA	Benchmark evaluating resistance to common falsehoods.
TruthX	Latent editing method that nudges activations toward truthful directions.
Tuned Lens	Learns linear mappings from hidden states to logits to study/steer layer-wise predictions.
TWEAK	Think While Effectively Articulating Knowledge—Hypothesis-and-NLI-guided reranking that prefers supported continuations.
U	
UHGEval	Hallucination evaluation benchmark for unconstrained generation in Chinese and related settings.
UPRISE	Uses LLM signals to train a retriever that selects stronger prompts/evidence.
V	
Verbose Cloning	Prompting/aggregation technique that elicits explicit, fully specified answers to reduce ambiguity.
X	
XCoT	Cross-lingual Chain-of-Thought prompting/transfer.
XNLI	Cross-lingual NLI benchmark commonly used for entailment-based verification.

Appendix A. Hallucination Mitigation Subcategories Comparison Table

Hallucination Mitigation Subcategories Comparison Table

Sub-Category	Scalability	Latency	Cost	Effectiveness	Implementation Complexity	Type of Method
1. Training and Learning Approaches						
a. Supervised and Semi-supervised Learning	Low	Low	High	High	High	Mature
b. Reinforcement Learning	Low	Low	High	High	High	Experimental
c. Contrastive Learning	Low	Low	High	High	High	Experimental
d. Knowledge Distillation	Low	Low	Medium	Medium	High	Experimental
e. Instruction Tuning	Low	Low	High	High	High	Experimental
2. Architectural Modifications						
a. Attention Mechanisms	High	Low	Low	Variable	High	Mature
b. Decoding Strategies	High	Low	Low	Variable	Medium	Mature
c. Retrieval Augmented Generation (RAG)	High	High	Medium	High	Medium	Mature
d. Knowledge Representation Approaches	Variable	Variable	Medium	High	High	Experimental
e. Specialized Architectural Mechanisms	Low	Low	High	High	High	Experimental
3. Input / Prompt Optimization						
a. Prompt Engineering	Variable	Low	Low	Variable	Low	Mature
b. Structured or Iterative Reasoning Prompting	Variable	Medium	Low	Variable	Low	Mature
c. In-context Prompting	Variable	Low	Low	Variable	Low	Experimental
d. Context Optimization	Variable	Low	Low	Variable	Low	Experimental
e. System Prompt Design	Variable	Low	Low	Variable	Low	Experimental
4. Post-Generation Quality Control						
a. Self-Verification and Consistency Checking	High	Medium	Low	Variable	Medium	Developing
b. External Fact-checking and Source Attribution	High	High	Medium	High	Medium	Mature
c. Uncertainty Estimation and Confidence Scoring	High	Low	Low	Variable	Medium	Developing
d. Output Refinement	High	Medium	Low	Variable	Medium	Mature
e. Response Validation	High	Medium	Low	Variable	Medium	Experimental
5. Interpretability and Diagnostic Approaches						
a. Internal State Probing	High	Low	Low	Low	Medium	Developing
b. Neuron Activation and Layer Analysis	High	Low	Low	Low	High	Developing
c. Attribution-based Diagnostics	High	Low	Low	Low	High	Experimental
6. Agent-based Orchestration						
a. Reflexive and Self-reflective Agents	Variable	High	Medium	Variable	High	Experimental
b. Modular and Multi-agent Architectures	Variable	High	Medium	Variable	High	Experimental

Legend

High · Medium · Low · Variable

Method Types (based on research volume):

Mature 30+ papers · Developing 20–30 papers · Experimental <20 papers

Appendix B. Summary Table of Benchmarks Used in Hallucination Detection and Mitigation

Factual Verification Benchmarks Category i

Benchmark	Brief Description
ANAH	Bilingual sentence-level hallucination labels (type + fix), used to train/evaluate detectors.
BoolQ	Yes/no RC over passages, used to test contextual understanding and binary factuality.
DiaHalu	Dialogue-level hallucination eval across four multi-turn domains, used for conversational robustness.
FACTOR	Factuality with multi-hop evidence retrieval, used for claim verification over multiple sources.
FACTSCORE	Atomic fact scoring with 6.5k generations, used to mark supported/unsupported/unverifiable units.
FELM	Long-form factuality eval across five domains, used to benchmark factuality evaluators.
FEVER	185k claims labeled Supported/Refuted/NEI, used for evidence retrieval + verification (often multi-hop).
FEWL	Label-free hallucination measurement via self-consistency + external checks, used without gold answers.
FRANK	Fine-grained error typology for summarization, used to compare factuality metrics on summaries.
HADES	Reference-free QA hallucination dataset, model-in-the-loop; used when ground truth is unavailable.
HalluEditBench	Verified hallucinations for model editing; scores efficacy, generalization, portability, locality, robustness.
HalluLens	Separates hallucination from factuality with intrinsic/extrinsic tasks, dynamic sets to reduce leakage.
HALOGEN	Nine-domain eval (code, attribution, history, reasoning, etc.), used for rate, refusals, and utility.
HaluEval	Large-scale cross-task benchmark for hallucination tendency and factual accuracy.
HaluEval 2.0	8,770 harder questions with stronger metrics, used to capture subtle hallucinations and robustness.
HaluEval-Wild	Adversarial real-user prompts (ShareGPT), used to test in-the-wild hallucinations.
HDMBench	Span/sentence annotations for knowledge-intensive tasks, used for fine-grained detection eval.
Head-to-Tail	18k QA by popularity (head/torso/tail), used to probe recall from common to rare knowledge.
HotpotQA	Multi-hop QA with supporting paragraphs, used to assess cross-document reasoning.
Natural Questions (NQ)	Real Google queries with Wikipedia answers, used for large-corpus retrieval + long-form QA.
RAGTruth	Word-level hallucination labels in RAG, used to benchmark hallucination frequency in pipelines.
SelfCheckGPT	Black-box self-consistency sampling, used to flag hallucinations without external DBs.
TriviaQA	650k QA-evidence triplets, used for diverse-topic QA with frequent multi-hop needs.
TruthfulQA	Adversarial questions targeting misconceptions, used to test truthfulness under lure-like claims.
UHGEval Chinese	Large unconstrained-gen dataset with rigorous annotation, used for realistic hallucination analysis.

Domain-Specific Benchmarks Category ii

Benchmark	Brief description
PubMedQA medical	Biomedical QA over literature, used to test domain factuality and clinical reasoning.
SciBench science	Scientific reasoning and claim checks, used to assess principle application and consistency.
LegalBench legal	Legal tasks for interpretation and reasoning, used to probe statutory/domain knowledge.

Code Generation Benchmarks Category iii

Benchmark	Brief Description
HumanEval / Codex	Function-level coding tasks with unit tests, used to measure correctness, reasoning, and API recall.

Figure A1. Summary Table of Benchmarks Used in Hallucination detection and mitigation research, detailing their source and key characteristics. The benchmarks summarized here are sourced from the following papers: [5,10,53,57,85,107,111,114,144,203,224,230,240,243,255,260,267,302,310–320].

References

1. Tonmoy, S.M.T.I.; Zaman, S.M.M.; Jain, V.; Rani, A.; Rawte, V.; Chadha, A.; Das, A. A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models. January 2024. Available online: http://arxiv.org/abs/2401.01313 (accessed on 12 August 2025).
2. Rawte, V.; Sheth, A.; Das, A. A Survey of Hallucination in Large Foundation Models. September 2023. Available online: http://arxiv.org/abs/2309.05922 (accessed on 12 August 2025).
3. Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Trans. Inf. Syst.* **2024**, *43*, 1–55. [CrossRef]
4. Agrawal, A.; Suzgun, M.; Mackey, L.; Kalai, A.T. Do Language Models Know When They're Hallucinating References? May 2023. Available online: http://arxiv.org/abs/2305.18248 (accessed on 12 August 2025).
5. Li, J.; Chen, J.; Ren, R.; Cheng, X.; Zhao, W.X.; Nie, J.; Wen, J. The Dawn After the Dark: An Empirical Study on Factuality Hallucination in Large Language Models. January 2024. Available online: http://arxiv.org/abs/2401.03205 (accessed on 12 August 2025).
6. Garcia-Carmona, A.M.; Prieto, M.-L.; Puertas, E.; Beunza, J.-J. Enhanced Medical Data Extraction: Leveraging LLMs for Accurate Retrieval of Patient Information from Medical Reports. *JMIR AI*. November 2024. Available online: https://www.researchgate.net/publication/382224134_Enhanced_Medical_Data_Extraction_Leveraging_LLMs_for_Accurate_Retrieval_of_Patient_Information_from_Medical_Reports (accessed on 12 August 2025).
7. Kim, Y.; Jeong, H.; Chen, S.; Li, S.S.; Lu, M.; Alhamoud, K.; Mun, J.; Grau, C.; Jung, M.; Gameiro, R.; et al. Medical Hallucinations in Foundation Models and Their Impact on Healthcare. February 2025. Available online: http://arxiv.org/abs/2503.05777 (accessed on 12 August 2025).
8. Magesh, V.; Surani, F.; Dahl, M.; Suzgun, M.; Manning, C.D.; Ho, D.E. Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools. May 2024. Available online: http://arxiv.org/abs/2405.20362 (accessed on 12 August 2025).
9. Dahl, M.; Magesh, V.; Suzgun, M.; Ho, D.E. Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models. *J. Leg. Anal.* **2024**, *16*, 64–93. [CrossRef]
10. Guha, N.; Nyarko, J.; Ho, D.E.; Ré, C.; Chilton, A.; Narayana, A.; Chohlas-Wood, A.; Peters, A.; Waldon, B.; Rockmore, D.N.; et al. LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models. August 2023. Available online: http://arxiv.org/abs/2308.11462 (accessed on 12 August 2025).
11. Shrivastava, A.; Hullman, J.; Lamparth, M. Measuring Free-Form Decision-Making Inconsistency of Language Models in Military Crisis Simulations. October 2024. Available online: http://arxiv.org/abs/2410.13204 (accessed on 12 August 2025).
12. Kalai, A.T.; Vempala, S.S. Calibrated Language Models Must Hallucinate. November 2023. Available online: http://arxiv.org/abs/2311.14648 (accessed on 12 August 2025).
13. Xu, Z.; Jain, S.; Kankanhalli, M. Hallucination is Inevitable: An Innate Limitation of Large Language Models. January 2024. Available online: http://arxiv.org/abs/2401.11817 (accessed on 12 August 2025).
14. Banerjee, S.; Agarwal, A.; Singla, S. LLMs Will Always Hallucinate, and We Need to Live with This. 2024. Available online: https://arxiv.org/abs/2409.05746 (accessed on 12 August 2025).
15. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training Language Models to Follow Instructions with Human Feedback. March 2022. Available online: http://arxiv.org/abs/2203.02155 (accessed on 12 August 2025).
16. Sun, W.; Shi, Z.; Gao, S.; Ren, P.; de Rijke, M.; Ren, Z. Contrastive Learning Reduces Hallucination in Conversations. December 2022. Available online: http://arxiv.org/abs/2212.10400 (accessed on 12 August 2025).
17. Li, X.L.; Holtzman, A.; Fried, D.; Liang, P.; Eisner, J.; Hashimoto, T.; Zettlemoyer, L.; Lewis, M. Contrastive Decoding: Open-ended Text Generation as Optimization. October 2022. Available online: http://arxiv.org/abs/2210.15097 (accessed on 12 August 2025).
18. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. May 2020. Available online: http://arxiv.org/abs/2005.11401 (accessed on 12 August 2025).
19. Thoppilan, R.; Freitas, D.D.; Hall, J.; Shazeer, N.; Kulshreshtha, A.; Cheng, H.; Jin, A.; Bos, T.; Baker, L.; Du, Y.; et al. LaMDA: Language Models for Dialog Applications. January 2022. Available online: http://arxiv.org/abs/2201.08239 (accessed on 12 August 2025).
20. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. January 2022. Available online: http://arxiv.org/abs/2201.11903 (accessed on 12 August 2025).
21. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.H.; Narang, S.; Chowdhery, A.; Zhou, D. Self-Consistency Improves Chain of Thought Reasoning in Language Models. May 2023. Available online: https://arxiv.org/abs/2203.11171 (accessed on 12 August 2025).

22. Li, M.; Peng, B.; Galley, M.; Gao, J.; Zhang, Z. Self-Checker: Plug-and-Play Modules for Fact-Checking with Large Language Models. May 2023. Available online: http://arxiv.org/abs/2305.14623 (accessed on 12 August 2025).
23. Welleck, S.; Lu, X.; West, P.; Brahman, F.; Shen, T.; Khashabi, D.; Choi, Y. Generating Sequences by Learning to Self-Correct. October 2022. Available online: http://arxiv.org/abs/2211.00053 (accessed on 12 August 2025).
24. Leiser, F.; Eckhardt, S.; Knaeble, M.; Maedche, A.; Schwabe, G.; Sunyaev, A. From ChatGPT to FactGPT: A Participatory Design Study to Mitigate the Effects of Large Language Model Hallucinations on Users. In Proceedings of the ACM International Conference Proceeding Series, New York, NY, USA, 26 September 2023; Association for Computing Machinery: New York, NY, USA, 2023; pp. 81–90. [CrossRef]
25. Farquhar, S.; Kossen, J.; Kuhn, L.; Gal, Y. Detecting hallucinations in large language models using semantic entropy. *Nature* **2024**, *630*, 625–630. [CrossRef]
26. Chen, C.; Liu, K.; Chen, Z.; Gu, Y.; Wu, Y.; Tao, M.; Fu, Z.; Ye, J. INSIDE: LLMs' Internal States Retain the Power of Hallucination Detection. February 2024. Available online: http://arxiv.org/abs/2402.03744 (accessed on 12 August 2025).
27. Bilal, A.; Mohsin, M.A.; Umer, M.; Bangash, M.A.K.; Jamshed, M.A. Meta-Thinking in LLMs via Multi-Agent Reinforcement Learning: A Survey. April 2025. Available online: http://arxiv.org/abs/2504.14520 (accessed on 12 August 2025).
28. Shinn, N.; Cassano, F.; Berman, E.; Gopinath, A.; Narasimhan, K.; Yao, S. Reflexion: Language Agents with Verbal Reinforcement Learning. March 2023. Available online: http://arxiv.org/abs/2303.11366 (accessed on 12 August 2025).
29. Xu, Z. Context-Aware Decoding Reduces Hallucination in Query-Focused Summarization. December 2023. Available online: http://arxiv.org/abs/2312.14335 (accessed on 12 August 2025).
30. Slobodkin, A.; Goldman, O.; Caciularu, A.; Dagan, I.; Ravfogel, S. The Curious Case of Hallucinatory (Un)Answerability: Finding Truths in the Hidden States of Over-Confident Large Language Models. October 2023. Available online: http://arxiv.org/abs/2310.11877 (accessed on 12 August 2025).
31. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.; Chen, D.; Dai, W.; et al. Survey of Hallucination in Natural Language Generation. July 2024. Available online: https://arxiv.org/pdf/2202.03629 (accessed on 12 August 2025).
32. Berberette, E.; Hutchins, J.; Sadovnik, A. Redefining 'Hallucination' in LLMs: Towards a Psychology-Informed Framework for Mitigating Misinformation. February 2024. Available online: http://arxiv.org/abs/2402.01769 (accessed on 12 August 2025).
33. van Deemter, K. The Pitfalls of Defining Hallucination. January 2024. Available online: http://arxiv.org/abs/2401.07897 (accessed on 12 August 2025).
34. Lee, K.; Ippolito, D.; Nystrom, A.; Zhang, C.; Eck, D.; Callison-Burch, C.; Carlini, N. Deduplicating Training Data Makes Language Models Better. July 2021. Available online: http://arxiv.org/abs/2107.06499 (accessed on 12 August 2025).
35. Carlini, N.; Ippolito, D.; Jagielski, M.; Lee, K.; Tramer, F.; Zhang, C. Quantifying Memorization Across Neural Language Models. February 2022. Available online: http://arxiv.org/abs/2202.07646 (accessed on 12 August 2025).
36. McKenna, N.; Li, T.; Cheng, L.; Hosseini, M.J.; Johnson, M.; Steedman, M. Sources of Hallucination by Large Language Models on Inference Tasks. May 2023. Available online: http://arxiv.org/abs/2305.14552 (accessed on 12 August 2025).
37. Lin, Z.; Guan, S.; Zhang, W.; Zhang, H.; Li, Y.; Zhang, H. Towards Trustworthy LLMs: A Review on Debiasing and Dehallucinating in Large Language Models. *Artif. Intell. Rev.* **2024**, *57*, 243. [CrossRef]
38. Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; Casas, D.d.L.; Hendricks, L.A.; Welbl, J.; Clark, A.; et al. Training Compute-Optimal Large Language Models. March 2022. Available online: http://arxiv.org/abs/2203.15556 (accessed on 12 August 2025).
39. Dziri, N.; Milton, S.; Yu, M.; Zaiane, O.; Reddy, S. On the Origin of Hallucinations in Conversational Models: Is it the Datasets or the Models? April 2022. Available online: http://arxiv.org/abs/2204.07931 (accessed on 12 August 2025).
40. Li, J.; Consul, S.; Zhou, E.; Wong, J.; Farooqui, N.; Ye, Y.; Manohar, N.; Wei, Z.; Wu, T.; Echols, B.; et al. Banishing LLM Hallucinations Requires Rethinking Generalization. June 2024. Available online: http://arxiv.org/abs/2406.17642 (accessed on 12 August 2025).
41. Yao, J.-Y.; Ning, K.-P.; Liu, Z.-H.; Ning, M.-N.; Liu, Y.-Y.; Yuan, L. LLM Lies: Hallucinations Are Not Bugs, but Features as Adversarial Examples. October 2023. Available online: http://arxiv.org/abs/2310.01469 (accessed on 12 August 2025).
42. Köksal, A.; Aksitov, R.; Chang, C.-C. Hallucination Augmented Recitations for Language Models. November 2023. Available online: http://arxiv.org/abs/2311.07424 (accessed on 12 August 2025).
43. He, Z.; Zhang, B.; Cheng, L. Shakespearean Sparks: The Dance of Hallucination and Creativity in LLMs' Decoding Layers. March 2025. Available online: http://arxiv.org/abs/2503.02851 (accessed on 12 August 2025).
44. Gundogmusler, A.; Bayindiroglu, F.; Karakucukoglu, M. Mathematical Foundations of Hallucination in Transformer-Based Large Language Models for Improvisation. *TechRxiv* **2024**. [CrossRef] [PubMed]
45. Ji, Z.; Yu, T.; Xu, Y.; Lee, N.; Ishii, E.; Fung, P. Towards Mitigating Hallucination in Large Language Models via Self-Reflection. October 2023. Available online: http://arxiv.org/abs/2310.06271 (accessed on 12 August 2025).
46. McIntosh, T.R.; Liu, T.; Susnjak, T.; Watters, P.; Ng, A.; Halgamuge, M.N. A Culturally Sensitive Test to Evaluate Nuanced GPT Hallucination. *IEEE Access* **2024**, *12*, 51555–51572. [CrossRef]

47. Shah, S.V. Accuracy, Consistency, and Hallucination of Large Language Models When Analyzing Unstructured Clinical Notes in Electronic Medical Records. *JAMA Netw. Open* **2024**, *7*, e2425953. [CrossRef]
48. Maleki, N.; Padmanabhan, B.; Dutta, K. AI Hallucinations: A Misnomer Worth Clarifying. January 2024. Available online: http://arxiv.org/abs/2401.06796 (accessed on 12 August 2025).
49. Yin, Z. A review of methods for alleviating hallucination issues in large language models. *Appl. Comput. Eng.* **2024**, *76*, 258–266. [CrossRef]
50. Ye, H.; Liu, T.; Zhang, A.; Hua, W.; Jia, W. Cognitive Mirage: A Review of Hallucinations in Large Language Models. September 2023. Available online: http://arxiv.org/abs/2309.06794 (accessed on 12 August 2025).
51. Zhang, W.; Zhang, J. Hallucination Mitigation for Retrieval-Augmented Large Language Models: A Review. *Mathematics* **2025**, *13*, 856. [CrossRef]
52. Perković, G.; Drobnjak, A.; Botički, I. Hallucinations in LLMs: Understanding and Addressing Challenges. In Proceedings of the 2024 47th ICT and Electronics Convention, MIPRO 2024-Proceedings, Opatija, Croatia, 20–24 May 2024; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2024; pp. 2084–2088. [CrossRef]
53. Lin, S.; Hilton, J.; Evans, O. TruthfulQA: Measuring How Models Mimic Human Falsehoods. Long Papers. May 2022. Available online: https://arxiv.org/abs/2109.07958 (accessed on 12 August 2025).
54. Zhang, Y.; Li, S.; Liu, J.; Yu, P.; Fung, Y.R.; Li, J.; Li, M.; Ji, H. Knowledge Overshadowing Causes Amalgamated Hallucination in Large Language Models. July 2024. Available online: http://arxiv.org/abs/2407.08039 (accessed on 12 August 2025).
55. Su, W.; Tang, Y.; Ai, Q.; Wang, C.; Wu, Z.; Liu, Y. Mitigating Entity-Level Hallucination in Large Language Models. In Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, Washington DC, USA, 14–18 December 2024; ACM: New York, NY, USA, 2024; pp. 23–31. [CrossRef]
56. Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; Fu, T.; Huang, X.; Zhao, E.; Zhang, Y.; Chen, Y.; et al. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. September 2023. Available online: http://arxiv.org/abs/2309.01219 (accessed on 12 August 2025).
57. Ji, Z.; Gu, Y.; Zhang, W.; Lyu, C.; Lin, D.; Chen, K. ANAH: Analytical Annotation of Hallucinations in Large Language Models. May 2024. Available online: http://arxiv.org/abs/2405.20315 (accessed on 12 August 2025).
58. Maynez, J.; Narayan, S.; Bohnet, B.; McDonald, R. On Faithfulness and Factuality in Abstractive Summarization. May 2020. Available online: http://arxiv.org/abs/2005.00661 (accessed on 12 August 2025).
59. Rawte, V.; Chakraborty, S.; Pathak, A.; Sarkar, A.; Tonmoy, S.M.T.I.; Chadha, A.; Sheth, A.P.; Das, A. The Troubling Emergence of Hallucination in Large Language Models—An Extensive Definition, Quantification, and Prescriptive Remediations. October 2023. Available online: http://arxiv.org/abs/2310.04988 (accessed on 12 August 2025).
60. Nan, F.; Nallapati, R.; Wang, Z.; Santos CNd Zhu, H.; Zhang, D.; McKeown, K.; Xiang, B. Entity-level Factual Consistency of Abstractive Text Summarization. February 2021. Available online: http://arxiv.org/abs/2102.09130 (accessed on 12 August 2025).
61. Guan, X.; Liu, Y.; Lin, H.; Lu, Y.; He, B.; Han, X.; Sun, L. Mitigating Large Language Model Hallucinations via Autonomous Knowledge Graph-Based Retrofitting. 2024. Available online: http://arxiv.org/abs/2311.13314 (accessed on 12 August 2025).
62. Vu, T.; Iyyer, M.; Wang, X.; Constant, N.; Wei, J.; Wei, J.; Tar, C.; Sung, Y.; Zhou, D.; Le, Q. FreshLLMs: Refreshing Large Language Models with Search Engine Augmentation. October 2023. Available online: http://arxiv.org/abs/2310.03214 (accessed on 12 August 2025).
63. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. Emergent Abilities of Large Language Models. June 2022. Available online: http://arxiv.org/abs/2206.07682 (accessed on 12 August 2025).
64. Besta, M.; Blach, N.; Kubicek, A.; Gerstenberger, R.; Podstawski, M.; Gianinazzi, L.; Gajda, J.; Lehmann, T.; Niewiadomski, H.; Nyczyk, P.; et al. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proc. AAAI Conf. Artif. Intell.* **2023**, *38*, 17682–17690. [CrossRef]
65. Zhang, M.; Press, O.; Merrill, W.; Liu, A.; Smith, N.A. How Language Model Hallucinations Can Snowball. May 2023. Available online: http://arxiv.org/abs/2305.13534 (accessed on 12 August 2025).
66. Zhang, Z.; Wang, Y.; Wang, C.; Chen, J.; Zheng, Z. LLM Hallucinations in Practical Code Generation: Phenomena, Mechanism, and Mitigation. September 2024. Available online: http://arxiv.org/abs/2409.20550 (accessed on 12 August 2025).
67. Mündler, N.; He, J.; Jenko, S.; Vechev, M. Self-Contradictory Hallucinations of Large Language Models: Evaluation, Detection and Mitigation. May 2023. Available online: http://arxiv.org/abs/2305.15852 (accessed on 12 August 2025).
68. Li, Y.; Li, Z.; Hung, K.; Wang, W.; Xie, H.; Li, Y. Ambiguity processing in Large Language Models: Detection, resolution, and the path to hallucination. *Nat. Lang. Process. J.* **2025**, 100173. [CrossRef]
69. Sharma, M.; Tong, M.; Korbak, T.; Duvenaud, D.; Askell, A.; Bowman, S.R.; Cheng, N.; Durmus, E.; Hatfield-Dodds, Z.; Johnston, S.R.; et al. Towards Understanding Sycophancy in Language Models. October 2023. Available online: http://arxiv.org/abs/2310.13548 (accessed on 12 August 2025).
70. Turpin, M.; Michael, J.; Perez, E.; Bowman, S.R. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. May 2023. Available online: http://arxiv.org/abs/2305.04388 (accessed on 12 August 2025).

71. Si, C.; Gan, Z.; Yang, Z.; Wang, S.; Wang, J.; Boyd-Graber, J.; Wang, L. Prompting GPT-3 To Be Reliable. October 2022. Available online: http://arxiv.org/abs/2210.09150 (accessed on 12 August 2025).
72. Zamfirescu-Pereira, J.D.; Wong, R.Y.; Hartmann, B.; Yang, Q. Why Johnny Can't Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts. In Proceedings of the Conference on Human Factors in Computing Systems-Proceedings, Association for Computing Machinery, Hamburg, Germany, 23–28 April 2023. [CrossRef]
73. Gao, T.; Fisch, A.; Chen, D. Making Pre-trained Language Models Better Few-Shot Learners. June 2021. Available online: http://arxiv.org/abs/2012.15723 (accessed on 12 August 2025).
74. Shuster, K.; Poff, S.; Chen, M.; Kiela, D.; Weston, J. Retrieval Augmentation Reduces Hallucination in Conversation. April 2021. Available online: http://arxiv.org/abs/2104.07567 (accessed on 12 August 2025).
75. Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; Choi, Y. The Curious Case of Neural Text Degeneration. April 2019. Available online: http://arxiv.org/abs/1904.09751 (accessed on 12 August 2025).
76. Tian, R.; Narayan, S.; Sellam, T.; Parikh, A.P. Sticking to the Facts: Confident Decoding for Faithful Data-to-Text Generation. October 2019. Available online: http://arxiv.org/abs/1910.08684 (accessed on 12 August 2025).
77. Xiong, M.; Hu, Z.; Lu, X.; Li, Y.; Fu, J.; He, J.; Hooi, B. Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs. June 2023. Available online: http://arxiv.org/abs/2306.13063 (accessed on 12 August 2025).
78. Qiu, X.; Miikkulainen, R. Semantic Density: Uncertainty Quantification for Large Language Models Through Confidence Measurement in Semantic Space. May 2024. Available online: http://arxiv.org/abs/2405.13845 (accessed on 12 August 2025).
79. Simhi, A.; Itzhak, I.; Barez, F.; Stanovsky, G.; Me, Y.B.T. I'm Wrong: High-Certainty Hallucinations in LLMs. February 2025. Available online: http://arxiv.org/abs/2502.12964 (accessed on 12 August 2025).
80. Schick, T.; Udupa, S.; Schütze, H. Self-Diagnosis and Self-Debiasing: A Proposal for Reducing Corpus-Based Bias in NLP. February 2021. Available online: http://arxiv.org/abs/2103.00453 (accessed on 12 August 2025).
81. Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. Constitutional AI: Harmlessness from AI Feedback. December 2022. Available online: http://arxiv.org/abs/2212.08073 (accessed on 12 August 2025).
82. Jha, S.; Jha, S.K.; Lincoln, P.; Bastian, N.D.; Velasquez, A.; Neema, S. Dehallucinating Large Language Models Using Formal Methods Guided Iterative Prompting. In Proceedings of the 2023 IEEE International Conference on Assured Autonomy, ICAA 2023, Laurel, MD, USA, 6–8 June 2023; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2023; pp. 149–152. [CrossRef]
83. Azaria, A.; Mitchell, T. The Internal State of an LLM Knows When It's Lying. April 2023. Available online: http://arxiv.org/abs/2304.13734 (accessed on 12 August 2025).
84. Luo, J.; Xiao, C.; Ma, F. Zero-Resource Hallucination Prevention for Large Language Models. September 2023. Available online: http://arxiv.org/abs/2309.02654 (accessed on 12 August 2025).
85. Luo, J.; Li, T.; Wu, D.; Jenkin, M.; Liu, S.; Dudek, G. Hallucination Detection and Hallucination Mitigation: An Investigation. January 2024. Available online: http://arxiv.org/abs/2401.08358 (accessed on 12 August 2025).
86. Liu, F.; Liu, Y.; Shi, L.; Huang, H.; Wang, R.; Yang, Z.; Zhang, L.; Li, Z.; Ma, Y. Exploring and Evaluating Hallucinations in LLM-Powered Code Generation. April 2024. Available online: http://arxiv.org/abs/2404.00971 (accessed on 12 August 2025).
87. Zhao, Y.; Liu, Z.; Zheng, Y.; Lam, K.-Y. Attribution Techniques for Mitigating Hallucination in RAG-based Question-Answering Systems: A Survey. *TechRxiv* **2025**. [CrossRef]
88. Agrawal, G.; Kumarage, T.; Alghamdi, Z.; Liu, H. Can Knowledge Graphs Reduce Hallucinations in LLMs? A Survey. March 2024. Available online: http://arxiv.org/abs/2311.07914 (accessed on 12 August 2025).
89. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. May 2020. Available online: http://arxiv.org/abs/2005.14165 (accessed on 12 August 2025).
90. Li, K.; Zhang, Y.; Li, K.; Fu, Y. Adversarial Feature Hallucination Networks for Few-Shot Learning. 2020. Available online: http://arxiv.org/abs/2003.13193 (accessed on 12 August 2025).
91. Filippova, K. Controlled Hallucinations: Learning to Generate Faithfully from Noisy Data. October 2020. Available online: http://arxiv.org/abs/2010.05873 (accessed on 12 August 2025).
92. Wan, D.; Bansal, M. FactPEGASUS: Factuality-Aware Pre-training and Fine-tuning for Abstractive Summarization. May 2022. Available online: http://arxiv.org/abs/2205.07830 (accessed on 12 August 2025).
93. Mishra, A.; Asai, A.; Balachandran, V.; Wang, Y.; Neubig, G.; Tsvetkov, Y.; Hajishirzi, H. Fine-grained Hallucination Detection and Editing for Language Models. January 2024. Available online: http://arxiv.org/abs/2401.06855 (accessed on 12 August 2025).
94. Hu, M.; He, B.; Wang, Y.; Li, L.; Ma, C.; King, I. Mitigating Large Language Model Hallucination with Faithful Finetuning. June 2024. Available online: http://arxiv.org/abs/2406.11267 (accessed on 12 August 2025).

95. Sun, Z.; Shen, Y.; Zhou, Q.; Zhang, H.; Chen, Z.; Cox, D.; Yang, Y.; Gan, C. Principle-Driven Self-Alignment of Language Models from Scratch with Minimal Human Supervision. May 2023. Available online: http://arxiv.org/abs/2305.03047 (accessed on 12 August 2025).
96. Zhou, C.; Neubig, G.; Gu, J.; Diab, M.; Guzmán, F.; Zettlemoyer, L.; Ghazvininejad, M. Detecting Hallucinated Content in Conditional Neural Sequence Generation. 2021. Available online: https://arxiv.org/abs/2011.02593 (accessed on 12 August 2025).
97. Chen, A.; Pasupat, P.; Singh, S.; Lee, H.; Guu, K. PURR: Efficiently Editing Language Model Hallucinations by Denoising Language Model Corruptions. May 2023. Available online: http://arxiv.org/abs/2305.14908 (accessed on 12 August 2025).
98. Gekhman, Z.; Herzig, J.; Aharoni, R.; Elkind, C.; Szpektor, I. TrueTeacher: Learning Factual Consistency Evaluation with Large Language Models. May 2023. Available online: http://arxiv.org/abs/2305.11171 (accessed on 12 August 2025).
99. Hu, Y.; Gan, L.; Xiao, W.; Kuang, K.; Wu, F. Fine-tuning Large Language Models for Improving Factuality in Legal Question Answering. January 2025. Available online: http://arxiv.org/abs/2501.06521 (accessed on 12 August 2025).
100. Qiu, Y.; Ziser, Y.; Korhonen, A.; Ponti, E.M.; Cohen, S.B. Detecting and Mitigating Hallucinations in Multilingual Summarisation. May 2023. Available online: http://arxiv.org/abs/2305.13632 (accessed on 12 August 2025).
101. Tang, Z.; Chatterjee, R.; Garg, S. Mitigating Hallucinated Translations in Large Language Models with Hallucination-focused Preference Optimization. January 2025. Available online: http://arxiv.org/abs/2501.17295 (accessed on 12 August 2025).
102. Cheng, D.; Huang, S.; Bi, J.; Zhan, Y.; Liu, J.; Wang, Y.; Sun, H.; Wei, F.; Deng, D.; Zhang, Q. UPRISE: Universal Prompt Retrieval for Improving Zero-Shot Evaluation. March 2023. Available online: http://arxiv.org/abs/2303.08518 (accessed on 12 August 2025).
103. Razumovskaia, E.; Vulić, I.; Marković, P.; Cichy, T.; Zheng, Q.; Wen, T.; Budzianowski, P. Dial BEINFO for Faithfulness: Improving Factuality of Information-Seeking Dialogue via Behavioural Fine-Tuning. November 2023. Available online: http://arxiv.org/abs/2311.09800 (accessed on 12 August 2025).
104. Elaraby, M.; Lu, M.; Dunn, J.; Zhang, X.; Wang, Y.; Liu, S.; Tian, P.; Wang, Y.; Wang, Y. Halo: Estimation and Reduction of Hallucinations in Open-Source Weak Large Language Models. August 2023. Available online: http://arxiv.org/abs/2308.11764 (accessed on 12 August 2025).
105. Rehman, T.; Mandal, R.; Agarwal, A.; Sanyal, D.K. Hallucination Reduction in Long Input Text Summarization. September 2023. Available online: http://arxiv.org/abs/2309.16781 (accessed on 12 August 2025).
106. Gekhman, Z.; Yona, G.; Aharoni, R.; Eyal, M.; Feder, A.; Reichart, R.; Herzig, J. Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations? May 2024. Available online: http://arxiv.org/abs/2405.05904 (accessed on 12 August 2025).
107. Zhu, Z.; Yang, Y.; Sun, Z. HaluEval-Wild: Evaluating Hallucinations of Language Models in the Wild. March 2024. Available online: http://arxiv.org/abs/2403.04307 (accessed on 12 August 2025).
108. Xia, Y.; Liu, X.; Yu, T.; Kim, S.; Rossi, R.A.; Rao, A.; Mai, T.; Li, S. Hallucination Diversity-Aware Active Learning for Text Summarization. April 2024. Available online: http://arxiv.org/abs/2404.01588 (accessed on 12 August 2025).
109. Cao, H.; An, Z.; Feng, J.; Xu, K.; Chen, L.; Zhao, D. A Step Closer to Comprehensive Answers: Constrained Multi-Stage Question Decomposition with Large Language Models. November 2023. Available online: http://arxiv.org/abs/2311.07491 (accessed on 12 August 2025).
110. Goodrich, B.; Rao, V.; Liu, P.J.; Saleh, M. Assessing the Factual Accuracy of Generated Text. In Proceedings of the KDD'19: The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019. [CrossRef]
111. Paudel, B.; Lyzhov, A.; Joshi, P.; Anand, P. HalluciNot: Hallucination Detection Through Context and Common Knowledge Verification. April 2025. Available online: http://arxiv.org/abs/2504.07069 (accessed on 12 August 2025).
112. Zou, A.; Phan, L.; Chen, S.; Campbell, J.; Guo, P.; Ren, R.; Pan, A.; Yin, X.; Mazeika, M.; Dombrowski, A.; et al. Representation Engineering: A Top-Down Approach to AI Transparency. October 2023. Available online: http://arxiv.org/abs/2310.01405 (accessed on 12 August 2025).
113. Li, R.; Luo, Z.; Du, X. FG-PRM: Fine-grained Hallucination Detection and Mitigation in Language Model Mathematical Reasoning. October 2024. Available online: http://arxiv.org/abs/2410.06304 (accessed on 12 August 2025).
114. Niu, C.; Wu, Y.; Zhu, J.; Xu, S.; Shum, K.; Zhong, R.; Song, J.; Zhang, T. RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models. December 2023. Available online: http://arxiv.org/abs/2401.00396 (accessed on 12 August 2025).
115. Lango, M.; Dušek, O. Critic-Driven Decoding for Mitigating Hallucinations in Data-to-text Generation. 2023. Available online: https://arxiv.org/abs/2310.16964 (accessed on 12 August 2025).
116. Lee, N.; Ping, W.; Xu, P.; Patwary, M.; Fung, P.; Shoeybi, M.; Catanzaro, B. Factuality Enhanced Language Models for Open-Ended Text Generation. June 2022. Available online: http://arxiv.org/abs/2206.04624 (accessed on 12 August 2025).
117. Pacchiardi, L.; Chan, A.J.; Mindermann, S.; Moscovitz, I.; Pan, A.Y.; Gal, Y.; Evans, O.; Brauner, J. How to Catch an AI Liar: Lie Detection in Black-Box LLMs by Asking Unrelated Questions. September 2023. Available online: http://arxiv.org/abs/2309.15840 (accessed on 12 August 2025).
118. Pfeiffer, J.; Piccinno, F.; Nicosia, M.; Wang, X.; Reid, M.; Ruder, S. mmT5: Modular Multilingual Pre-Training Solves Source Language Hallucinations. May 2023. Available online: http://arxiv.org/abs/2305.14224 (accessed on 12 August 2025).

119. Qiu, Y.; Embar, V.; Cohen, S.B.; Han, B. Think While You Write: Hypothesis Verification Promotes Faithful Knowledge-to-Text Generation. November 2023. Available online: http://arxiv.org/abs/2311.09467 (accessed on 12 August 2025).
120. Chen, Z.; Sun, X.; Jiao, X.; Lian, F.; Kang, Z.; Wang, D.; Xu, C. Truth Forest: Toward Multi-Scale Truthfulness in Large Language Models through Intervention without Tuning. December 2023. Available online: http://arxiv.org/abs/2312.17484 (accessed on 12 August 2025).
121. Zhang, S.; Yu, T.; Feng, Y. TruthX: Alleviating Hallucinations by Editing Large Language Models in Truthful Space. February 2024. Available online: http://arxiv.org/abs/2402.17811 (accessed on 12 August 2025).
122. Lewis, A.; White, M.; Liu, J.; Koike-Akino, T.; Parsons, K.; Wang, Y. Winning Big with Small Models: Knowledge Distillation vs. Self-Training for Reducing Hallucination in QA Agents. February 2025. Available online: http://arxiv.org/abs/2502.19545 (accessed on 12 August 2025).
123. Xu, C.; Sun, Q.; Zheng, K.; Geng, X.; Zhao, P.; Feng, J.; Tao, C.; Jiang, D. WizardLM: Empowering Large Language Models to Follow Complex Instructions. April 2023. Available online: http://arxiv.org/abs/2304.12244 (accessed on 12 August 2025).
124. Longpre, S.; Perisetla, K.; Chen, A.; Nikhil, R.; Dubois, C.; Singh, S. Entity-Based Knowledge Conflicts in Question Answering. 2021. Available online: https://arxiv.org/abs/2109.05052 (accessed on 3 August 2025).
125. Sutton, R.S.; Barto, A.G. Reinforcement Learning: An Introduction. 2015. Available online: https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf (accessed on 12 August 2025).
126. Roit, P.; Ferret, J.; Shani, L.; Aharoni, R.; Cideron, G.; Dadashi, R.; Geist, M.; Girgin, S.; Hussenot, L.; Keller, O.; et al. Factually Consistent Summarization via Reinforcement Learning with Textual Entailment Feedback. May 2023. Available online: http://arxiv.org/abs/2306.00186 (accessed on 12 August 2025).
127. Tian, K.; Mitchell, E.; Yao, H.; Manning, C.D.; Finn, C. Fine-Tuning Language Models for Factuality. November 2023. Available online: http://arxiv.org/abs/2311.08401 (accessed on 12 August 2025).
128. Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; Cobbe, K. Let's Verify Step by Step. May 2023. Available online: http://arxiv.org/abs/2305.20050 (accessed on 12 August 2025).
129. Christiano, P.; Leike, J.; Brown, T.B.; Martic, M.; Legg, S.; Amodei, D. Deep Reinforcement Learning from Human Preferences. June 2017. Available online: http://arxiv.org/abs/1706.03741 (accessed on 12 August 2025).
130. Ji, J.; Qiu, T.; Chen, B.; Zhang, B.; Lou, H.; Wang, K.; Duan, Y.; He, Z.; Vierling, L.; Hong, D.; et al. AI Alignment: A Comprehensive Survey. October 2023. Available online: http://arxiv.org/abs/2310.19852 (accessed on 12 August 2025).
131. OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; et al. *GPT-4 Technical Report*. March 2023. Available online: http://arxiv.org/abs/2303.08774 (accessed on 12 August 2025).
132. Yang, Y.; Chern, E.; Qiu, X.; Neubig, G.; Liu, P. Alignment for Honesty. October 2024. Available online: http://arxiv.org/abs/2312.07000 (accessed on 12 August 2025).
133. Perez, E.; Ringer, S.; Lukošiūtė, K.; Nguyen, K.; Chen, E.; Heiner, S.; Pettit, C.; Olsson, C.; Kundu, S.; Kadavath, S.; et al. Discovering Language Model Behaviors with Model-Written Evaluations. December 2022. Available online: http://arxiv.org/abs/2212.09251 (accessed on 12 August 2025).
134. Lee, H.; Phatale, S.; Mansoor, H.; Mesnard, T.; Ferret, J.; Lu, K.; Bishop, C.; Hall, E.; Carbune, V.; Rastogi, A.; et al. RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. September 2023. Available online: http://arxiv.org/abs/2309.00267 (accessed on 12 August 2025).
135. Liang, Y.; Song, Z.; Wang, H.; Zhang, J. Learning to Trust Your Feelings: Leveraging Self-awareness in LLMs for Hallucination Mitigation. January 2024. Available online: http://arxiv.org/abs/2401.15449 (accessed on 12 August 2025).
136. Cheng, X.; Li, J.; Zhao, W.X.; More, J.-R.W.T. Hallucinate Less: Mitigating Hallucinations via Dual Process of Fast and Slow Thinking. January 2025. Available online: http://arxiv.org/abs/2501.01306 (accessed on 12 August 2025).
137. Lin, S.; Gao, L.; Oguz, B.; Xiong, W.; Lin, J.; Yih, W.; Chen, X. FLAME: Factuality-Aware Alignment for Large Language Models. May 2024. Available online: http://arxiv.org/abs/2405.01525 (accessed on 12 August 2025).
138. Parcalabescu, L.; Frank, A. On Measuring Faithfulness or Self-consistency of Natural Language Explanations. November 2023. Available online: http://arxiv.org/abs/2311.07466 (accessed on 12 August 2025).
139. Gosmar, D.; Dahl, D.A. Hallucination Mitigation Using Agentic AI Natural Language-Based Frameworks. January 2025. Available online: http://arxiv.org/abs/2501.13946 (accessed on 12 August 2025).
140. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. February 2020. Available online: http://arxiv.org/abs/2002.05709 (accessed on 12 August 2025).
141. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum Contrast for Unsupervised Visual Representation Learning. November 2019. Available online: http://arxiv.org/abs/1911.05722 (accessed on 12 August 2025).
142. Chern, I.-C.; Wang, Z.; Das, S.; Sharma, B.; Liu, P.; Neubig, G. Improving Factuality of Abstractive Summarization via Contrastive Reward Learning. July 2023. Available online: http://arxiv.org/abs/2307.04507 (accessed on 12 August 2025).
143. Robinson, J.; Chuang, C.-Y.; Sra, S.; Jegelka, S. Contrastive Learning with Hard Negative Samples. October 2020. Available online: http://arxiv.org/abs/2010.04592 (accessed on 12 August 2025).

144. Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.W.; Salakhutdinov, R.; Manning, C.D. HOTPOTQA: A Dataset for Diverse, Explainable Multi-Hop Question Answering. Association for Computational Linguistics. Available online: https://arxiv.org/abs/1809.09600 (accessed on 12 August 2025).
145. Wu, H.; Li, X.; Xu, X.; Wu, J.; Zhang, D.; Liu, Z. Iter-AHMCL: Alleviate Hallucination for Large Language Model via Iterative Model-level Contrastive Learning. October 2024. Available online: http://arxiv.org/abs/2410.12130 (accessed on 12 August 2025).
146. Gema, A.P.; Jin, C.; Abdulaal, A.; Diethe, T.; Teare, P.; Alex, B.; Minervini, P.; Saseendran, A. DeCoRe: Decoding by Contrasting Retrieval Heads to Mitigate Hallucinations. October 2024. Available online: http://arxiv.org/abs/2410.18860 (accessed on 12 August 2025).
147. Huang, C.P.; Chen, H.-Y. Delta—Contrastive Decoding Mitigates Text Hallucinations in Large Language Models. February 2025. Available online: http://arxiv.org/abs/2502.05825 (accessed on 12 August 2025).
148. He, J.; Gong, Y.; Chen, K.; Lin, Z.; Wei, C.; Zhao, Y. LLM Factoscope: Uncovering LLMs' Factual Discernment through Inner States Analysis. December 2023. Available online: http://arxiv.org/abs/2312.16374 (accessed on 12 August 2025).
149. Wang, P.; Wang, Z.; Li, Z.; Gao, Y.; Yin, B.; Ren, X. SCOTT: Self-Consistent Chain-of-Thought Distillation. 2023. Available online: http://arxiv.org/abs/2305.01879 (accessed on 12 August 2025).
150. Chuang, Y.-S.; Xie, Y.; Luo, H.; Kim, Y.; Glass, J.; He, P. DoLa: Decoding by Contrasting Layers Improves Factuality in Large Language Models. September 2023. Available online: http://arxiv.org/abs/2309.03883 (accessed on 12 August 2025).
151. Xu, W.; Agrawal, S.; Briakou, E.; Martindale, M.J.; Carpuat, M. Understanding and Detecting Hallucinations in Neural Machine Translation via Model Introspection. January 2023. Available online: http://arxiv.org/abs/2301.07779 (accessed on 12 August 2025).
152. Nguyen, H.; He, Z.; Gandre, S.A.; Pasupulety, U.; Shivakumar, S.K.; Lerman, K. Smoothing Out Hallucinations: Mitigating LLM Hallucination with Smoothed Knowledge Distillation. February 2025. Available online: http://arxiv.org/abs/2502.11306 (accessed on 12 August 2025).
153. Liu, W.; Li, G.; Zhang, K.; Du, B.; Chen, Q.; Hu, X.; Xu, H.; Chen, J.; Wu, J. Mind's Mirror: Distilling Self-Evaluation Capability and Comprehensive Thinking from Large Language Models. November 2023. Available online: http://arxiv.org/abs/2311.09214 (accessed on 12 August 2025).
154. Feng, J.; Wang, Q.; Qiu, H.; Liu, L. Retrieval In Decoder benefits generative models for explainable complex question answering. *Neural Netw.* **2025**, *181*, 106833. [CrossRef] [PubMed]
155. Zhang, H.; Diao, S.; Lin, Y.; Fung, Y.R.; Lian, Q.; Wang, X.; Chen, Y.; Ji, H.; Zhang, T. R-Tuning: Instructing Large Language Models to Say 'I Don't Know'. November 2023. Available online: http://arxiv.org/abs/2311.09677 (accessed on 12 August 2025).
156. Chung, H.W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. Scaling Instruction-Finetuned Language Models. October 2022. Available online: http://arxiv.org/abs/2210.11416 (accessed on 12 August 2025).
157. Wan, F.; Huang, X.; Cui, L.; Quan, X.; Bi, W.; Shi, S. Knowledge Verification to Nip Hallucination in the Bud. January 2024. Available online: http://arxiv.org/abs/2401.10768 (accessed on 12 August 2025).
158. Zhao, Y.; Yan, L.; Sun, W.; Xing, G.; Wang, S.; Meng, C.; Cheng, Z.; Ren, Z.; Yin, D. Improving the Robustness of Large Language Models via Consistency Alignment. March 2024. Available online: http://arxiv.org/abs/2403.14221 (accessed on 12 August 2025).
159. Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N.A.; Khashabi, D.; Hajishirzi, H. Self-Instruct: Aligning Language Models with Self-Generated Instructions. December 2022. Available online: http://arxiv.org/abs/2212.10560 (accessed on 12 August 2025).
160. Zheng, W.; Lee, R.K.-W.; Liu, Z.; Wu, K.; Aw, A.; Zou, B. CCL-XCoT: An Efficient Cross-Lingual Knowledge Transfer Method for Mitigating Hallucination Generation. July 2025. Available online: http://arxiv.org/abs/2507.14239 (accessed on 12 August 2025).
161. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. September 2014. Available online: http://arxiv.org/abs/1409.0473 (accessed on 12 August 2025).
162. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. June 2017. Available online: http://arxiv.org/abs/1706.03762 (accessed on 12 August 2025).
163. Michel, P.; Levy, O.; Neubig, G. Are Sixteen Heads Really Better than One? November 2019. Available online: http://arxiv.org/abs/1905.10650 (accessed on 12 August 2025).
164. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.-J. BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318. [CrossRef]
165. Li, K.; Liu, T.; Bashkansky, N.; Bau, D.; Viégas, F.; Pfister, H.; Wattenberg, M. Measuring and Controlling Instruction (In)Stability in Language Model Dialogs. February 2024. Available online: http://arxiv.org/abs/2402.10962 (accessed on 12 August 2025).

166. Hoscilowicz, J.; Wiacek, A.; Chojnacki, J.; Cieslak, A.; Michon, L.; Urbanevych, V.; Janicki, A. Non-Linear Inference Time Intervention: Improving LLM Truthfulness. March 2024. Available online: http://arxiv.org/abs/2403.18680 (accessed on 12 August 2025).
167. Fairburn, S.; Ainsworth, J. Mitigate Large Language Model Hallucinations with Probabilistic Inference in Graph Neural Networks. 1 July 2024. Available online: https://www.authorea.com/users/798018/articles/1147827-mitigate-large-language-model-hallucinations-with-probabilistic-inference-in-graph-neural-networks?commit=59e46cef9e4db14a5daf553bcbf96ff7ebab29be (accessed on 12 August 2025).
168. Shelmanov, A.; Fadeeva, E.; Tsvigun, A.; Tsvigun, I.; Xie, Z.; Kiselev, I.; Daheim, N.; Zhang, C.; Vazhentsev, A.; Sachan, M.; et al. A Head to Predict and a Head to Question: Pre-trained Uncertainty Quantification Heads for Hallucination Detection in LLM Outputs. May 2025. Available online: http://arxiv.org/abs/2505.08200 (accessed on 12 August 2025).
169. Guo, L.; Fang, Y.; Chen, F.; Liu, P.; Xu, S. Large Language Models with Adaptive Token Fusion: A Novel Approach to Reducing Hallucinations and Improving Inference Efficiency. 24 October 2024. Available online: https://www.authorea.com/users/847419/articles/1235237-large-language-models-with-adaptive-token-fusion-a-novel-approach-to-reducing-hallucinations-and-improving-inference-efficiency?commit=8e85c59f4f49cf8895c0b1eb937d89a716932d4c (accessed on 12 August 2025).
170. Yuksekgonul, M.; Chandrasekaran, V.; Jones, E.; Gunasekar, S.; Naik, R.; Palangi, H.; Kamar, E.; Nushi, B. Attention Satisfies: A Constraint-Satisfaction Lens on Factual Errors of Language Models. September 2023. Available online: http://arxiv.org/abs/2309.15098 (accessed on 12 August 2025).
171. Nie, F.; Yao, J.-G.; Wang, J.; Pan, R.; Lin, C.-Y. A Simple Recipe towards Reducing Hallucination in Neural Surface Realisation. Association for Computational Linguistics. 2019. Available online: https://aclanthology.org/P19-1256.pdf (accessed on 12 August 2025).
172. Matys, P.; Eliasz, J.; Kiełczyński, K.; Langner, M.; Ferdinan, T.; Kocoń, J.; Kazienko, P. AggTruth: Contextual Hallucination Detection using Aggregated Attention Scores in LLMs. June 2025. [CrossRef]
173. Shi, W.; Han, X.; Lewis, M.; Tsvetkov, Y.; Zettlemoyer, L.; Yih, S.W. Trusting Your Evidence: Hallucinate Less with Context-aware Decoding. May 2023. Available online: http://arxiv.org/abs/2305.14739 (accessed on 12 August 2025).
174. Wu, J.; Shen, Y.; Liu, S.; Tang, Y.; Song, S.; Wang, X.; Cai, L. Improve Decoding Factuality by Token-wise Cross Layer Entropy of Large Language Models. 2025. Available online: https://arxiv.org/abs/2502.03199 (accessed on 12 August 2025).
175. van der Poel, L.; Cotterell, R.; Meister, C. Mutual Information Alleviates Hallucinations in Abstractive Summarization. October 2022. Available online: http://arxiv.org/abs/2210.13210 (accessed on 12 August 2025).
176. Shi, W.; Min, S.; Yasunaga, M.; Seo, M.; James, R.; Lewis, M.; Zettlemoyer, L.; Yih, W. REPLUG: Retrieval-Augmented Black-Box Language Models. January 2023. Available online: http://arxiv.org/abs/2301.12652 (accessed on 12 August 2025).
177. Xiao, Y.; Wang, W.Y. On Hallucination and Predictive Uncertainty in Conditional Language Generation. March 2021. Available online: http://arxiv.org/abs/2103.15025 (accessed on 12 August 2025).
178. Huang, L.; Feng, X.; Ma, W.; Fan, Y.; Feng, X.; Gu, Y.; Ye, Y.; Zhao, L.; Zhong, W.; Wang, B.; et al. Alleviating Hallucinations from Knowledge Misalignment in Large Language Models via Selective Abstention Learning. 2025. Available online: https://aclanthology.org/2025.acl-long.1199.pdf (accessed on 4 August 2025).
179. Kai, J.; Zhang, T.; Hu, H.; Lin, Z. SH2: Self-Highlighted Hesitation Helps You Decode More Truthfully. January 2024. Available online: http://arxiv.org/abs/2401.05930 (accessed on 12 August 2025).
180. Qiu, Y.; Zhao, Z.; Ziser, Y.; Korhonen, A.; Ponti, E.M.; Cohen, S.B. Spectral Editing of Activations for Large Language Model Alignment. May 2024. Available online: http://arxiv.org/abs/2405.09719 (accessed on 12 August 2025).
181. Zhang, Y.; Cui, L.; Bi, W.; Shi, S. Alleviating Hallucinations of Large Language Models through Induced Hallucinations. December 2023. Available online: http://arxiv.org/abs/2312.15710 (accessed on 12 August 2025).
182. Zhang, H.; Chen, H.; Chen, M.; Zhang, T. Active Layer-Contrastive Decoding Reduces Hallucination in Large Language Model Generation. June 2025. Available online: http://arxiv.org/abs/2505.23657 (accessed on 12 August 2025).
183. Ji, Z.; Liu, Z.; Lee, N.; Yu, T.; Wilie, B.; Zeng, M.; Fung, P. RHO (ρ): Reducing Hallucination in Open-Domain Dialogues with Knowledge Grounding. May 2023. Available online: https://arxiv.org/abs/2212.01588 (accessed on 12 August 2025).
184. Chen, S.; Xiong, M.; Liu, J.; Wu, Z.; Xiao, T.; Gao, S.; He, J. In-Context Sharpness as Alerts: An Inner Representation Perspective for Hallucination Mitigation. March 2024. Available online: http://arxiv.org/abs/2403.01548 (accessed on 12 August 2025).
185. Li, K.; Patel, O.; Viégas, F.; Pfister, H.; Wattenberg, M. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. June 2023. Available online: http://arxiv.org/abs/2306.03341 (accessed on 12 August 2025).
186. Elhoushi, M.; Shrivastava, A.; Liskovich, D.; Hosmer, B.; Wasti, B.; Lai, L.; Mahmoud, A.; Acun, B.; Agrawal, S.; Roman, A.; et al. LayerSkip: Enabling Early Exit Inference and Self-Speculative Decoding. 2024. Available online: https://arxiv.org/abs/2404.16710 (accessed on 3 August 2025).
187. Chen, J.; Lin, H.; Han, X.; Sun, L. Benchmarking Large Language Models in Retrieval-Augmented Generation. September 2023. Available online: http://arxiv.org/abs/2309.01431 (accessed on 12 August 2025).

188. Dziri, N.; Madotto, A.; Zaiane, O.; Bose, A.J. Neural Path Hunter: Reducing Hallucination in Dialogue Systems via Path Grounding. April 2021. Available online: http://arxiv.org/abs/2104.08455 (accessed on 12 August 2025).
189. Yu, H.Q.; McQuade, F. RAG-KG-IL: A Multi-Agent Hybrid Framework for Reducing Hallucinations and Enhancing LLM Reasoning Through RAG and Incremental Knowledge Graph Learning Integration. March 2025. Available online: http://arxiv.org/abs/2503.13514 (accessed on 12 August 2025).
190. Gao, L.; Dai, Z.; Pasupat, P.; Chen, A.; Chaganty, A.T.; Fan, Y.; Zhao, V.Y.; Lao, N.; Lee, H.; Juan, D.; et al. RARR: Researching and Revising What Language Models Say, Using Language Models. October 2022. Available online: http://arxiv.org/abs/2210.08726 (accessed on 12 August 2025).
191. Karpukhin, V.; Guz, B.O.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; Yih, W.; Ai, F. Dense Passage Retrieval for Open-Domain Question Answering. 2020. Available online: https://arxiv.org/abs/2004.04906 (accessed on 12 August 2025).
192. Mala, C.S.; Gezici, G.; Giannotti, F. Hybrid Retrieval for Hallucination Mitigation in Large Language Models: A Comparative Analysis. February 2025. Available online: http://arxiv.org/abs/2504.05324 (accessed on 12 August 2025).
193. Peng, B.; Galley, M.; He, P.; Cheng, H.; Xie, Y.; Hu, Y.; Huang, Q.; Liden, L.; Yu, Z.; Chen, W.; et al. Check Your Facts and Try Again: Improving Large Language Models with External Knowledge and Automated Feedback. February 2023. Available online: http://arxiv.org/abs/2302.12813 (accessed on 12 August 2025).
194. CH-Wang, S.; van Durme, B.; Eisner, J.; Kedzie, C. Do Androids Know They're Only Dreaming of Electric Sheep? December 2023. Available online: http://arxiv.org/abs/2312.17249 (accessed on 12 August 2025).
195. Barry, M.; Caillaut, G.; Halftermeyer, P.; Qader, R.; Mouayad, M.; Cariolaro, D.; Deit, F.L.; Gesnouin, J. GraphRAG: Leveraging Graph-Based Efficiency to Minimize Hallucinations in LLM-Driven RAG for Finance Data. 2025. Available online: https://aclanthology.org/2025.genaik-1.6.pdf (accessed on 12 August 2025).
196. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Self-RAG, H.H. Generate, and Critique through Self-Reflection. October 2023. Available online: http://arxiv.org/abs/2310.11511 (accessed on 12 August 2025).
197. Dwivedi, K.; Mishra, P.P. AutoRAG-LoRA: Hallucination-Triggered Knowledge Retuning via Lightweight Adapters. July 2025. Available online: http://arxiv.org/abs/2507.10586 (accessed on 12 August 2025).
198. Cao, S.; Zhang, J.; Shi, J.; Lv, X.; Yao, Z.; Tian, Q.; Li, J.; Hou, L. Probabilistic Tree-of-thought Reasoning for Answering Knowledge-intensive Complex Questions. November 2023. Available online: http://arxiv.org/abs/2311.13982 (accessed on 12 August 2025).
199. Su, D.; Li, X.; Zhang, J.; Shang, L.; Jiang, X.; Liu, Q.; Fung, P. Read before Generate! Faithful Long Form Question Answering with Machine Reading. 2022. Available online: https://arxiv.org/abs/2203.00343 (accessed on 12 May 2025).
200. Signé, Q.; Boughanem, M.; Moreno, J.G.; Belkacem, T. A Substring Extraction-Based RAG Method for Minimising Hallucinations in Aircraft Maintenance Question Answering. In Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR), New York, NY, USA, 18 July 2025; ACM: New York, NY, USA, 2025; pp. 513–521. [CrossRef]
201. Lv, Q.; Wang, J.; Chen, H.; Li, B.; Zhang, Y.; Wu, F. Coarse-to-Fine Highlighting: Reducing Knowledge Hallucination in Large Language Models. October 2024. Available online: http://arxiv.org/abs/2410.15116 (accessed on 12 August 2025).
202. Nonkes, N.; Agaronian, S.; Kanoulas, E.; Petcu, R. Leveraging Graph Structures to Detect Hallucinations in Large Language Models. July 2024. Available online: http://arxiv.org/abs/2407.04485 (accessed on 12 August 2025).
203. Sun, K.; Xu, Y.E.; Zha, H.; Liu, Y.; Dong, X.L. Head-to-Tail: How Knowledgeable are Large Language Models (LLMs)? A.K.A. Will LLMs Replace Knowledge Graphs? August 2023. Available online: http://arxiv.org/abs/2308.10168 (accessed on 12 August 2025).
204. Lavrinovics, E.; Biswas, R.; Bjerva, J.; Hose, K.K. Knowledge Graphs, Large Language Models, and Hallucinations: An NLP Perspective. 2024. Available online: http://arxiv.org/abs/2411.14258 (accessed on 12 August 2025).
205. Zhang, S.; Pan, L.; Zhao, J.; Wang, W.Y. The Knowledge Alignment Problem: Bridging Human and External Knowledge for Large Language Models. May 2023. Available online: http://arxiv.org/abs/2305.13669 (accessed on 12 August 2025).
206. Reddy, G.P.; Kumar, Y.V.P.; Prakash, K.P. Hallucinations in Large Language Models (LLMs). In Proceedings of the 2024 IEEE Open Conference of Electrical, Electronic and Information Sciences, Vilnius, Lithuania, 25 April 2024; eStream 2024-Proceedings. Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2024. [CrossRef]
207. Bayat, F.F.; Qian, K.; Han, B.; Sang, Y.; Belyi, A.; Khorshidi, S.; Wu, F.; Ilyas, I.F.; Li, Y. FLEEK: Factual Error Detection and Correction with Evidence Retrieved from External Knowledge. October 2023. Available online: http://arxiv.org/abs/2310.17119 (accessed on 12 August 2025).
208. Sherif, S.; Saad, D.; Silva, S.; Gomes, V. Graph-Enhanced RAG: A Survey of Methods, Architectures, and Performance. 2025. Available online: https://www.researchgate.net/publication/393193258 (accessed on 12 August 2025).
209. Li, H.; Appleby, G.; Alperin, K.; Gomez, S.R.; Suh, A. Mitigating LLM Hallucinations with Knowledge Graphs: A Case Study. April 2025. Available online: http://arxiv.org/abs/2504.12422 (accessed on 12 August 2025).

210. Nishat, N.A.Z.; Coletta, A.; Bellomarini, L.; Amouzouvi, K.; Lehmann, J.; Vahdati, S. Aligning Knowledge Graphs and Language Models for Factual Accuracy. July 2025. Available online: http://arxiv.org/abs/2507.13411 (accessed on 12 August 2025).
211. He, X.; Tian, Y.; Sun, Y.; Chawla, N.V.; Laurent, T.; LeCun, Y.; Bresson, X.; Hooi, B. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. May 2024. Available online: http://arxiv.org/abs/2402.07630 (accessed on 12 August 2025).
212. Suzuoki, S.; Hatano, K. Reducing Hallucinations in Large Language Models: A Consensus Voting Approach Using Mixture of Experts. *TechRxiv* **2024**. [CrossRef]
213. Behore, S.; Dumont, L.; Venkataraman, J. Enhancing Reliability in Large Language Models: Self-Detection of Hallucinations with Spontaneous Self-Checks. 9 September 2024. Available online: https://www.authorea.com/users/829447/articles/1223513-enhancing-reliability-in-large-language-models-self-detection-of-hallucinations-with-spontaneous-self-checks?commit=5c3caaa663d1123b079882ae7501d480e3831a68 (accessed on 12 August 2025).
214. Chrysostomou, G.; Zhao, Z.; Williams, M.; Aletras, N. Investigating Hallucinations in Pruned Large Language Models for Abstractive Summarization. November 2023. Available online: https://arxiv.org/pdf/2311.09335 (accessed on 12 August 2025).
215. Jacobs, R.A.; Jordan, M.I.; Nowlan, S.J.; Hinton, G.E. Adaptive Mixtures of Local Experts. 1991. Available online: https://2024.sci-hub.se/1867/e922caa86bf169b2dbb314f150dbdadb/jacobs1991.pdf (accessed on 7 August 2025).
216. Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; Dean, J. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. January 2017. Available online: http://arxiv.org/abs/1701.06538 (accessed on 12 August 2025).
217. Li, J.; Mao, Z.; Wang, Q. Alleviating Hallucinations in Large Language Models via Truthfulness-Driven Rank-adaptive LoRA. July 2025. Available online: https://aclanthology.org/2025.findings-acl.103.pdf (accessed on 4 August 2025).
218. Wang, C.; Zhao, Y.; Liu, Y.; Zhu, H. Enhancing Latent Diffusion in Large Language Models for High-Quality Implicit Neural Representations with Reduced Hallucinations. 2024. Available online: https://osf.io/preprints/osf/9utwy_v1 (accessed on 29 June 2025).
219. Feldman, P.; Foulds, J.R.; Pan, S. Trapping LLM Hallucinations Using Tagged Context Prompts. June 2023. Available online: http://arxiv.org/abs/2306.06085 (accessed on 12 August 2025).
220. Lei, D.; Li, Y.; Hu, M.; Wang, M.; Yun, V.; Ching, E.; Kamal, E. Chain of Natural Language Inference for Reducing Large Language Model Ungrounded Hallucinations. October 2023. Available online: http://arxiv.org/abs/2310.03951 (accessed on 12 August 2025).
221. White, J.; Fu, Q.; Hays, S.; Sandborn, M.; Olea, C.; Gilbert, H.; Elnashar, A.; Spencer-Smith, J.; Schmidt, D.C. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. February 2023. Available online: http://arxiv.org/abs/2302.11382 (accessed on 12 August 2025).
222. Kaddour, J.; Harris, J.; Mozes, M.; Bradley, H.; Raileanu, R.; McHardy, R. Challenges and Applications of Large Language Models. July 2023. Available online: http://arxiv.org/abs/2307.10169 (accessed on 12 August 2025).
223. Cheng, Q.; Sun, T.; Zhang, W.; Wang, S.; Liu, X.; Zhang, M.; He, J.; Huang, M.; Yin, Z.; Chen, K.; et al. Evaluating Hallucinations in Chinese Large Language Models. October 2023. Available online: http://arxiv.org/abs/2310.03368 (accessed on 12 August 2025).
224. Li, J.; Cheng, X.; Zhao, W.X.; Nie, J.-Y.; Wen, J.-R. HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models. May 2023. Available online: http://arxiv.org/abs/2305.11747 (accessed on 12 August 2025).
225. Leiser, F.; Eckhardt, S.; Leuthe, V.; Knaeble, M.; Maedche, A.; Schwabe, G.; Sunyaev, A. HILL: A Hallucination Identifier for Large Language Models. In Proceedings of the Conference on Human Factors in Computing Systems-Proceedings, Association for Computing Machinery, New York, NY, USA, 11–16 May 2024. [CrossRef]
226. Levinstein, B.A.; Herrmann, D.A. Still No Lie Detector for Language Models: Probing Empirical and Conceptual Roadblocks. *Philos. Stud.* **2023**, *182*, 1539–1565. [CrossRef]
227. Varshney, N.; Raj, S.; Mishra, V.; Chatterjee, A.; Sarkar, R.; Saeidi, A.; Baral, C. Investigating and Addressing Hallucinations of LLMs in Tasks Involving Negation. June 2024. Available online: http://arxiv.org/abs/2406.05494 (accessed on 12 August 2025).
228. Cao, Z.; Yang, Y.; Zhao, H. AutoHall: Automated Hallucination Dataset Generation for Large Language Models. September 2023. Available online: http://arxiv.org/abs/2310.00259 (accessed on 12 August 2025).
229. Agarwal, V.; Pei, Y.; Alamir, S.; Liu, X. CodeMirage: Hallucinations in Code Generated by Large Language Models. August 2024. Available online: http://arxiv.org/abs/2408.08333 (accessed on 12 August 2025).
230. Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H.P.d.O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. Evaluating Large Language Models Trained on Code. July 2021. Available online: http://arxiv.org/abs/2107.03374 (accessed on 12 August 2025).
231. Hu, X.; Ru, D.; Qiu, L.; Guo, Q.; Zhang, T.; Xu, Y.; Luo, Y.; Liu, P.; Zhang, Y.; Zhang, Z. RefChecker: Reference-Based Fine-grained Hallucination Checker and Benchmark for Large Language Models. May 2024. Available online: http://arxiv.org/abs/2405.14486 (accessed on 12 August 2025).

232. Elchafei, P.; Abu-Elkheir, M. Span-Level Hallucination Detection for LLM-Generated Answers. April 2025. Available online: http://arxiv.org/abs/2504.18639 (accessed on 12 August 2025).
233. Hao, S.; Gu, Y.; Ma, H.; Hong, J.J.; Wang, Z.; Wang, D.Z.; Hu, Z. Reasoning with Language Model is Planning with World Model. May 2023. Available online: http://arxiv.org/abs/2305.14992 (accessed on 12 August 2025).
234. Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.L.; Cao, Y.; Narasimhan, K. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. May 2023. Available online: http://arxiv.org/abs/2305.10601 (accessed on 12 August 2025).
235. Lester, B.; Al-Rfou, R.; Constant, N.; Research, G. The Power of Scale for Parameter-Efficient Prompt Tuning. Available online: https://arxiv.org/abs/2104.08691 (accessed on 12 August 2025).
236. Liu, Y.; Deng, G.; Xu, Z.; Li, Y.; Zheng, Y.; Zhang, Y.; Zhao, L.; Zhang, T.; Wang, K.; Liu, Y. Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study. May 2023. Available online: http://arxiv.org/abs/2305.13860 (accessed on 12 August 2025).
237. Dhuliawala, S.; Komeili, M.; Xu, J.; Raileanu, R.; Li, X.; Celikyilmaz, A.; Weston, J. Chain-of-Verification Reduces Hallucination in Large Language Models. September 2023. Available online: http://arxiv.org/abs/2309.11495 (accessed on 12 August 2025).
238. Braverman, A.; Zhang, W.; Gu, Q. Mitigating Hallucination in Large Language Models with Explanatory Prompting. 2024. Available online: https://neurips.cc/virtual/2024/105546 (accessed on 12 August 2025).
239. Kıcıman, E.; Ness, R.; Sharma, A.; Tan, C. Causal Reasoning and Large Language Models: Opening a New Frontier for Causality. April 2023. Available online: http://arxiv.org/abs/2305.00050 (accessed on 12 August 2025).
240. Jin, Q.; Dhingra, B.; Liu, Z.; Cohen, W.W.; Lu, X. PubMedQA: A Dataset for Biomedical Research Question Answering. September 2019. Available online: http://arxiv.org/abs/1909.06146 (accessed on 12 August 2025).
241. Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; et al. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. May 2022. Available online: http://arxiv.org/abs/2205.10625 (accessed on 12 August 2025).
242. Yan, J.N.; Liu, T.; Chiu, J.T.; Shen, J.; Qin, Z.; Yu, Y.; Zhao, Y.; Lakshmanan, C.; Kurzion, Y.; Rush, A.M.; et al. Predicting Text Preference Via Structured Comparative Reasoning. November 2023. Available online: http://arxiv.org/abs/2311.08390 (accessed on 12 August 2025).
243. Wei, J.; Yao, Y.; Ton, J.-F.; Guo, H.; Estornell, A.; Liu, Y. Measuring and Reducing LLM Hallucination without Gold-Standard Answers. February 2024. Available online: http://arxiv.org/abs/2402.10412 (accessed on 12 August 2025).
244. Chern, I.; Chern, S.; Chen, S.; Yuan, W.; Feng, K.; Zhou, C.; He, J.; Neubig, G.; Liu, P. FacTool: Factuality Detection in Generative AI—A Tool Augmented Framework for Multi-Task and Multi-Domain Scenarios. July 2023. Available online: http://arxiv.org/abs/2307.13528 (accessed on 12 August 2025).
245. Li, N.; Li, Y.; Liu, Y.; Shi, L.; Wang, K.; Wang, H. Drowzee: Metamorphic Testing for Fact-Conflicting Hallucination Detection in Large Language Models. In *Proceedings of the ACM on Programming Languages*; ACM: New York, NY, USA, May 2024. [CrossRef]
246. Kang, H.; Ni, J.; Yao, H. Ever: Mitigating Hallucination in Large Language Models through Real-Time Verification and Rectification. November 2023. Available online: http://arxiv.org/abs/2311.09114 (accessed on 12 August 2025).
247. Yan, T.; Xu, T. Refining the Responses of LLMs by Themselves. May 2023. Available online: http://arxiv.org/abs/2305.04039 (accessed on 12 August 2025).
248. Du, L.; Wang, Y.; Xing, X.; Ya, Y.; Li, X.; Jiang, X.; Fang, X. Quantifying and Attributing the Hallucination of Large Language Models via Association Analysis. September 2023. Available online: http://arxiv.org/abs/2309.05217 (accessed on 12 August 2025).
249. Chang, E.Y. Prompting Large Language Models with the Socratic Method. February 2023. Available online: http://arxiv.org/abs/2303.08769 (accessed on 12 August 2025).
250. Yehuda, Y.; Malkiel, I.; Barkan, O.; Weill, J.; Ronen, R.; Koenigstein, N. InterrogateLLM: Zero-Resource Hallucination Detection in LLM-Generated Answers. August 2024. Available online: http://arxiv.org/abs/2403.02889 (accessed on 12 August 2025).
251. Cohen, R.; Hamri, M.; Geva, M.; Globerson, A. LM vs LM: Detecting Factual Errors via Cross Examination. May 2023. Available online: http://arxiv.org/abs/2305.13281 (accessed on 12 August 2025).
252. Kojima, T.; Gu, S.S.; Reid, M.; Matsuo, Y.; Iwasawa, Y. Large Language Models are Zero-Shot Reasoners. January 2023. Available online: http://arxiv.org/abs/2205.11916 (accessed on 12 August 2025).
253. Jones, E.; Palangi, H.; Simões, C.; Chandrasekaran, V.; Mukherjee, S.; Mitra, A.; Awadallah, A.; Kamar, E. Teaching Language Models to Hallucinate Less with Synthetic Tasks. October 2023. Available online: http://arxiv.org/abs/2310.06827 (accessed on 12 August 2025).
254. Zhao, T.Z.; Wallace, E.; Feng, S.; Klein, D.; Singh, S. Calibrate Before Use: Improving Few-Shot Performance of Language Models. June 2021. Available online: http://arxiv.org/abs/2102.09690 (accessed on 12 August 2025).
255. Min, S.; Krishna, K.; Lyu, X.; Lewis, M.; Yih, W.; Koh, P.W.; Iyyer, M.; Zettlemoyer, L.; Hajishirzi, H. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. May 2023. Available online: http://arxiv.org/abs/2305.14251 (accessed on 12 August 2025).
256. Gou, Z.; Shao, Z.; Gong, Y.; Shen, Y.; Yang, Y.; Duan, N.; Chen, W. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. May 2023. Available online: http://arxiv.org/abs/2305.11738 (accessed on 12 August 2025).

257. Suzgun, M.; Kalai, A.T. Meta-Prompting: Enhancing Language Models with Task-Agnostic Scaffolding. January 2024. Available online: http://arxiv.org/abs/2401.12954 (accessed on 12 August 2025).
258. Grayson, M.; Patterson, C.; Goldstein, B.; Ivanov, S.; Davidson, M. Mitigating Hallucinations in Large Language Models Using a Channel-Aware Domain-Adaptive Generative Adversarial Network (CADAGAN). 30 September 2024. Available online: https://www.researchsquare.com/article/rs-5164079/v1 (accessed on 12 August 2025).
259. Joshi, N.; Rando, J.; Saparov, A.; Kim, N.; He, H. Personas as a Way to Model Truthfulness in Language Models. October 2023. Available online: http://arxiv.org/abs/2310.18168 (accessed on 12 August 2025).
260. Chen, K.; Chen, Q.; Zhou, J.; He, Y.; He, L. DiaHalu: A Dialogue-level Hallucination Evaluation Benchmark for Large Language Models. March 2024. Available online: http://arxiv.org/abs/2403.00896 (accessed on 12 August 2025).
261. Xu, R.; Lin, B.S.; Yang, S.; Zhang, T.; Shi, W.; Zhang, T.; Fang, Z.; Xu, W.; Qiu, H. The Earth is Flat Because. . .: Investigating LLMs' Belief Towards Misinformation via Persuasive Conversation. December 2023. Available online: http://arxiv.org/abs/2312.09085 (accessed on 12 August 2025).
262. Chen, R.; Arditi, A.; Sleight, H.; Evans, O.; Lindsey, J. Persona Vectors: Monitoring and Controlling Character Traits in Language Models. July 2025. Available online: http://arxiv.org/abs/2507.21509 (accessed on 12 August 2025).
263. Li, X.L.; Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. January 2021. Available online: http://arxiv.org/abs/2101.00190 (accessed on 12 August 2025).
264. Kadavath, S.; Conerly, T.; Askell, A.; Henighan, T.; Drain, D.; Perez, E.; Schiefer, N.; Hatfield-Dodds, Z.; DasSarma, N.; Tran-Johnson, E.; et al. Language Models (Mostly) Know What They Know. November 2022. Available online: http://arxiv.org/abs/2207.05221 (accessed on 12 August 2025).
265. Wu, W.; Cao, Y.; Yi, N.; Ou, R.; Zheng, Z. Detecting and Reducing the Factual Hallucinations of Large Language Models with Metamorphic Testing. In Proceedings of the ACM on Software Engineering; ACM: New York, NY, USA, 2025; Volume 2, pp. 1432–1453. Available online: https://dl.acm.org/doi/pdf/10.1145/3715784 (accessed on 12 August 2025).
266. Harrington, F.; Rosenthal, E.; Swinburne, M. Mitigating Hallucinations in Large Language Models with Sliding Generation and Self-Checks. *TechRxiv* **2024**. [CrossRef]
267. Manakul, P.; Liusie, A.; Gales, M.J.F. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. March 2023. Available online: http://arxiv.org/abs/2303.08896 (accessed on 12 August 2025).
268. Zhao, R.; Li, X.; Joty, S.; Qin, C.; Bing, L. Verify-and-Edit: A Knowledge-Enhanced Chain-of-Thought Framework. May 2023. Available online: http://arxiv.org/abs/2305.03268 (accessed on 12 August 2025).
269. Zhao, Z.; Cohen, S.B.; Webber, B. Reducing Quantity Hallucinations in Abstractive Summarization. September 2020. Available online: http://arxiv.org/abs/2009.13312 (accessed on 12 August 2025).
270. Li, X.; Zhao, R.; Chia, Y.K.; Ding, B.; Joty, S.; Poria, S.; Bing, L. Chain-of-Knowledge: Grounding Large Language Models via Dynamic Knowledge Adapting over Heterogeneous Sources. May 2023. Available online: http://arxiv.org/abs/2305.13269 (accessed on 12 August 2025).
271. Zablocki, P.; Gajewska, Z. Assessing Hallucination Risks in Large Language Models Through Internal State Analysis. *Authorea* **2024**. [CrossRef]
272. Dale, D.; Voita, E.; Barrault, L.; Costa-jussà, M.R. Detecting and Mitigating Hallucinations in Machine Translation: Model Internal Workings Alone Do Well, Sentence Similarity Even Better. December 2022. Available online: http://arxiv.org/abs/2212.08597 (accessed on 12 August 2025).
273. Liu, Y.; Yang, Q.; Tang, J.; Guo, T.; Wang, C.; Li, P.; Xu, S.; Gao, X.; Li, Z.; Liu, J.; et al. Reducing hallucinations of large language models via hierarchical semantic piece. *Complex Intell. Syst.* **2025**, *11*, 231. [CrossRef]
274. Ross, J.J.; Khramtsova, E.; van der Vegt, A.; Koopman, B.; Zuccon, G. RARR Unraveled: Component-Level Insights into Hallucination Detection and Mitigation. In Proceedings of the SIGIR 2025, 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, Padua, Italy, 13–18 July 2025; Association for Computing Machinery, Inc.: New York, NY, USA, 2025; pp. 3286–3295. [CrossRef]
275. Kossen, J.; Han, J.; Razzak, M.; Schut, L.; Malik, S.; Gal, Y. Semantic Entropy Probes: Robust and Cheap Hallucination Detection in LLMs. 2024. Available online: https://arxiv.org/pdf/2406.15927 (accessed on 3 May 2025).
276. Verma, S.; Tran, K.; Ali, Y.; Min, G. Reducing LLM Hallucinations using Epistemic Neural Networks. December 2023. Available online: http://arxiv.org/abs/2312.15576 (accessed on 12 August 2025).
277. Yin, Z.; Sun, Q.; Guo, Q.; Wu, J.; Qiu, X.; Huang, X. Do Large Language Models Know What They Don't Know? 2023. Available online: https://github.com/yinzhangyue/SelfAware (accessed on 12 August 2025).
278. Lin, S.; Hilton, J.; Evans, O. Teaching Models to Express Their Uncertainty in Words. May 2022. Available online: http://arxiv.org/abs/2205.14334 (accessed on 12 August 2025).
279. Zhang, T.; Qiu, L.; Guo, Q.; Deng, C.; Zhang, Y.; Zhang, Z.; Zhou, C.; Wang, X.; Fu, L. Enhancing Uncertainty-Based Hallucination Detection with Stronger Focus. November 2023. Available online: http://arxiv.org/abs/2311.13230 (accessed on 12 August 2025).

280. Yan, S.-Q.; Gu, J.-C.; Zhu, Y.; Ling, Z.-H. Corrective Retrieval Augmented Generation. January 2024. Available online: http://arxiv.org/abs/2401.15884 (accessed on 12 August 2025).
281. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. Self-Refine: Iterative Refinement with Self-Feedback. March 2023. Available online: http://arxiv.org/abs/2303.17651 (accessed on 12 August 2025).
282. Guerreiro, N.M.; Voita, E.; Martins, A.F.T. Looking for a Needle in a Haystack: A Comprehensive Study of Hallucinations in Neural Machine Translation. August 2022. Available online: http://arxiv.org/abs/2208.05309 (accessed on 12 August 2025).
283. Alain, G.; Bengio, Y. Understanding Intermediate Layers Using Linear Classifier Probes. October 2016. Available online: http://arxiv.org/abs/1610.01644 (accessed on 12 August 2025).
284. Rateike, M.; Cintas, C.; Wamburu, J.; Akumu, T.; Speakman, S. Weakly Supervised Detection of Hallucinations in LLM Activations. December 2023. Available online: http://arxiv.org/abs/2312.02798 (accessed on 12 August 2025).
285. Chuang, Y.-S.; Qiu, L.; Hsieh, C.-Y.; Krishna, R.; Kim, Y.; Glass, J. Lookback Lens: Detecting and Mitigating Contextual Hallucinations in Large Language Models Using Only Attention Maps. October 2024. Available online: http://arxiv.org/abs/2407.07071 (accessed on 12 August 2025).
286. Zhu, D.; Chen, D.; Li, Q.; Chen, Z.; Ma, L.; Grossklags, J.; Fritz, M. PoLLMgraph: Unraveling Hallucinations in Large Language Models via State Transition Dynamics. April 2024. Available online: http://arxiv.org/abs/2404.04722 (accessed on 12 August 2025).
287. Marks, S.; Tegmark, M. The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets. October 2023. Available online: http://arxiv.org/abs/2310.06824 (accessed on 12 August 2025).
288. Chen, L.; Wu, X.; Xiong, Z.; Kang, X. Two Stage Psychology-Guided Fine-Grained Editing and Sampling Approach for Mitigating Hallucination in Large Language Models Publication. 2025. Available online: https://escholarship.org/uc/item/0gn8m1qq (accessed on 4 August 2025).
289. Son, M.; Jang, J.; Kim, M. Lightweight Query Checkpoint: Classifying Faulty User Queries to Mitigate Hallucinations in Large Language Model Question Answering. July 2025. Available online: https://openreview.net/pdf?id=n9C8u6tpT4 (accessed on 4 August 2025).
290. Yin, F.; Srinivasa, J.; Chang, K.-W. Characterizing Truthfulness in Large Language Model Generations with Local Intrinsic Dimension. February 2024. Available online: http://arxiv.org/abs/2402.18048 (accessed on 12 August 2025).
291. Dai, D.; Dong, L.; Hao, Y.; Sui, Z.; Chang, B.; Wei, F. Knowledge Neurons in Pretrained Transformers. March 2022. Available online: http://arxiv.org/abs/2104.08696 (accessed on 12 August 2025).
292. Raunak, V.; Menezes, A.; Junczys-Dowmunt, M. The Curious Case of Hallucinations in Neural Machine Translation. April 2021. Available online: http://arxiv.org/abs/2104.06683 (accessed on 12 August 2025).
293. Jiang, C.; Qi, B.; Hong, X.; Fu, D.; Cheng, Y.; Meng, F.; Yu, M.; Zhou, B.; Zhou, J. On Large Language Models' Hallucination with Regard to Known Facts. March 2024. Available online: http://arxiv.org/abs/2403.20009 (accessed on 12 August 2025).
294. Ji, Z.; Chen, D.; Ishii, E.; Cahyawijaya, S.; Bang, Y.; Wilie, B.; Fung, P. LLM Internal States Reveal Hallucination Risk Faced with a Query. July 2024. Available online: http://arxiv.org/abs/2407.03282 (accessed on 12 August 2025).
295. Yin, K.; Neubig, G. Interpreting Language Models with Contrastive Explanations. February 2022. Available online: http://arxiv.org/abs/2202.10419 (accessed on 12 August 2025).
296. Yu, L.; Cao, M.; Cheung, J.C.K.; Dong, Y. Mechanistic Understanding and Mitigation of Language Model Non-Factual Hallucinations. June 2024. Available online: http://arxiv.org/abs/2403.18167 (accessed on 12 August 2025).
297. Kapoor, S.; Stroebl, B.; Siegel, Z.S.; Nadgir, N.; Narayanan, A. AI Agents That Matter. July 2024. Available online: http://arxiv.org/abs/2407.01502 (accessed on 12 August 2025).
298. Thórisson, K.; Helgasson, H. Cognitive Architectures and Autonomy: A Comparative Review. *J. Artif. Gen. Intell.* **2012**, *3*, 1–30. [CrossRef]
299. Du, Y.; Li, S.; Torralba, A.; Tenenbaum, J.B.; Mordatch, I. Improving Factuality and Reasoning in Language Models through Multiagent Debate. May 2023. Available online: http://arxiv.org/abs/2305.14325 (accessed on 12 August 2025).
300. Abd Elrahman Amer and Magdi Amer Using Multi-Agent Architecture to Mitigate the Risk of LLM Hallucinations. July 2025. Available online: https://arxiv.org/pdf/2507.01446 (accessed on 5 July 2025).
301. Huh, D.; Mohapatra, P. Multi-Agent Reinforcement Learning: A Comprehensive Survey. July 2024. Available online: http://arxiv.org/abs/2312.10256 (accessed on 12 August 2025).
302. Pagnoni, A.; Balachandran, V.; Tsvetkov, Y. Understanding Factuality in Abstractive Summarization with FRANK: A Benchmark for Factuality Metrics. April 2021. Available online: http://arxiv.org/abs/2104.13346 (accessed on 12 August 2025).
303. Banerjee, S.; Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI, USA, 29 June 2005; pp. 65–72. Available online: https://aclanthology.org/W05-0909/ (accessed on 12 August 2025).

304. Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In Proceedings of the ACL Workshop on Text Summarization Branches Out, Barcelona, Spain, 25–26 July 2004; pp. 74–81.
305. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. April 2019. Available online: http://arxiv.org/abs/1904.09675 (accessed on 12 August 2025).
306. Chaturvedi, A.; Bhar, S.; Saha, S.; Garain, U.; Asher, N. Analyzing Semantic Faithfulness of Language Models via Input Intervention on Question Answering. *Comput. Linguist.* **2024**, *50*, 119–155. [CrossRef]
307. Kryściński, W.; McCann, B.; Xiong, C.; Socher, R. Evaluating the Factual Consistency of Abstractive Text Summarization. October 2019. Available online: http://arxiv.org/abs/1910.12840 (accessed on 12 August 2025).
308. Ramprasad, S.; Ferracane, E.; Lipton, Z.C. Analyzing LLM Behavior in Dialogue Summarization: Unveiling Circumstantial Hallucination Trends. June 2024. Available online: http://arxiv.org/abs/2406.03487 (accessed on 12 August 2025).
309. Hong, G.; Gema, A.P.; Saxena, R.; Du, X.; Nie, P.; Zhao, Y.; Perez-Beltrachini, L.; Ryabinin, M.; He, X.; Fourrier, C.; et al. The Hallucinations Leaderboard—An Open Effort to Measure Hallucinations in Large Language Models. April 2024. Available online: http://arxiv.org/abs/2404.05904 (accessed on 12 August 2025).
310. Clark, C.; Lee, K.; Chang, M.; Kwiatkowski, T.; Collins, M.; Toutanova, K.; Allen, P.G. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. Available online: https://arxiv.org/abs/1905.10044 (accessed on 12 August 2025).
311. Muhlgay, D.; Ram, O.; Magar, I.; Levine, Y.; Ratner, N.; Belinkov, Y.; Abend, O.; Leyton-Brown, K.; Shashua, A.; Shoham, Y. Generating Benchmarks for Factuality Evaluation of Language Models. July 2023. Available online: http://arxiv.org/abs/2307.06908 (accessed on 12 August 2025).
312. Chen, S.; Zhao, Y.; Zhang, J.; Chern, I.; Gao, S.; Liu, P.; He, J. FELM: Benchmarking Factuality Evaluation of Large Language Models. November 2023. Available online: http://arxiv.org/abs/2310.00741 (accessed on 12 August 2025).
313. Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; Mittal, A. FEVER: A Large-Scale Dataset for Fact Extraction and VERification. March 2018. Available online: http://arxiv.org/abs/1803.05355 (accessed on 12 August 2025).
314. Huang, B.; Chen, C.; Xu, X.; Payani, A.; Shu, K. Can Knowledge Editing Really Correct Hallucinations? March 2025. Available online: http://arxiv.org/abs/2410.16251 (accessed on 12 August 2025).
315. Bang, Y.; Ji, Z.; Schelten, A.; Hartshorn, A.; Fowler, T.; Zhang, C.; Cancedda, N.; Fung, P. HalluLens: LLM Hallucination Benchmark. April 2025. Available online: http://arxiv.org/abs/2504.17550 (accessed on 12 August 2025).
316. Ravichander, A.; Ghela, S.; Wadden, D.; Choi, Y. HALoGEN: Fantastic LLM Hallucinations and Where to Find Them. January 2025. Available online: http://arxiv.org/abs/2501.08292 (accessed on 12 August 2025).
317. Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. Natural Questions: A Benchmark for Question Answering Research. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 452–466. [CrossRef]
318. Joshi, M.; Choi, E.; Weld, D.S.; Zettlemoyer, L. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. May 2017. Available online: http://arxiv.org/abs/1705.03551 (accessed on 12 August 2025).
319. Liang, X.; Song, S.; Niu, S.; Li, Z.; Xiong, F.; Tang, B.; Wang, Y.; He, D.; Cheng, P.; Wang, Z.; et al. UHGEval: Benchmarking the Hallucination of Chinese Large Language Models via Unconstrained Generation. November 2023. Available online: https://huggingface.co/papers/2311.15296 (accessed on 12 August 2025).
320. Wang, X.; Hu, Z.; Lu, P.; Zhu, Y.; Zhang, J.; Subramaniam, S.; Loomba, A.R.; Zhang, S.; Sun, Y.; Wang, W. SciBench: Evaluating College-Level Scientific Problem-Solving Abilities of Large Language Models. July 2023. Available online: http://arxiv.org/abs/2307.10635 (accessed on 12 August 2025).
321. Chen, Y.; Fu, Q.; Yuan, Y.; Wen, Z.; Fan, G.; Liu, D.; Zhang, D.; Li, Z.; Xiao, Y. Hallucination Detection: Robustly Discerning Reliable Answers in Large Language Models. In Proceedings of the International Conference on Information and Knowledge Management, Birmingham, UK, 21–25 October 2023; Association for Computing Machinery: New York, NY, USA, 2023; pp. 245–255. Available online: https://arxiv.org/pdf/2407.04121 (accessed on 12 August 2025).
322. Laban, P.; Kryściński, W.; Agarwal, D.; Fabbri, A.R.; Xiong, C.; Joty, S.; Wu, C. LLMs as Factual Reasoners: Insights from Existing Benchmarks and Beyond. May 2023. Available online: http://arxiv.org/abs/2305.14540 (accessed on 12 August 2025).
323. Yoran, O.; Wolfson, T.; Ram, O.; Berant, J. Making Retrieval-Augmented Language Models Robust to Irrelevant Context. October 2023. Available online: http://arxiv.org/abs/2310.01558 (accessed on 12 August 2025).
324. Honovich, O.; Choshen, L.; Aharoni, R.; Neeman, E.; Szpektor, I.; Abend, O. Q2: Evaluating Factual Consistency in Knowledge-Grounded Dialogues via Question Generation and Question Answering. April 2021. Available online: http://arxiv.org/abs/2104.08202 (accessed on 12 August 2025).
325. Malin, B.; Kalganova, T.; Boulgouris, N. A Review of Faithfulness Metrics for Hallucination Assessment in Large Language Models. 2024. Available online: http://arxiv.org/abs/2501.00269 (accessed on 12 August 2025).
326. Huo, S.; Arabzadeh, N.; Clarke, C.L.A. Retrieving Supporting Evidence for LLMs Generated Answers. 2023. Available online: http://arxiv.org/abs/2306.13781 (accessed on 12 August 2025).
327. Naseem, T.; Xu, G.; Swaminathan, S.; Yehudai, A.; Chaudhury, S.; Florian, R.; Astudillo, R.; Munawar, A. A Grounded Preference Model for LLM Alignment. 2024. Available online: https://aclanthology.org/2024.findings-acl.10 (accessed on 12 August 2025).

328. Chen, E.; Kaushik, D.; Dhillon, G.; Wang, Y.; Hadsell, R.; Cohen, W.W. Walk the Talk? Measuring the Faithfulness of Large Language Model Explanations. 2025. Available online: https://arxiv.org/abs/2504.14150 (accessed on 12 August 2025).
329. Lanham, T.; Chen, A.; Radhakrishnan, A.; Steiner, B.; Denison, C.; Hernandez, D.; Li, D.; Durmus, E.; Hubinger, E.; Kernion, J.; et al. Measuring Faithfulness in Chain-of-Thought Reasoning. 2023. Available online: https://arxiv.org/abs/2307.13702 (accessed on 12 August 2025).
330. The PRISMA 2020 Statement: An Updated Guideline for Reporting Systematic Reviews. Available online: https://www.bmj.com/content/372/bmj.n71 (accessed on 26 September 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.