Review

Not peer-reviewed version

A Survey on Hallucination in Large Language Models: Definitions, Detection, and Mitigation

Seyed Mahmoud Sajjadi Mohammadabadi * , Burak Cem Kara , Can Eyupoglu , Oktay Karakus

Posted Date: 22 January 2026

doi: 10.20944/preprints202510.0540.v2

Keywords: hallucination; large language models (LLMs); factuality; faithfulness; hallucination detection; hallucination mitigation; multimodal AI; AI safety

 doi:10.20944/preprints202510.0540.v2

Review

A Survey on Hallucination in Large Language Models: Definitions, Detection, and Mitigation

Seyed Mahmoud Sajjadi Mohammadabadi [1], Burak Cem Kara [1], Can Eyupoglu [1,2] and Oktay Karakuş [3,*]

[1] University of Nevada, Reno
[2] Turkish Air Force Academy, National Defence University
[3] Cardiff University
* Correspondence: karakuso@cardiff.ac.uk

Abstract

Despite Large Language Models (LLMs) exhibiting outstanding capabilities in various natural language processing tasks, they might still be unreliable. Actually, one of their main sources of unreliability is a phenomenon called hallucination, the creation of reasonable but false pieces of information. This work provides a comprehensive overview of advances in understanding, locating, and reducing hallucinations. We start by considering hallucination as the main obstacle in creating reliable AI, and define a taxonomy that follows the development of factual errors and the notion of unfaithfulness with respect to the model's accessible knowledge. Afterwards, we survey the detection methods that are classified depending on the degree of model access and also, and we also refer to the different cognitive processes used for their comparison, which comprise uncertainty estimation, consistency checking, and knowledge-grounding evaluation. In the end, we offer a well-organized representation of the interventions aimed at the abolition of the model hallucinations employed at various stages of the model lifecycle: (1) data-centric interventions exemplified by high-quality data curation, (2) model-centric alignment through preference optimization and knowledge editing, and (3) inference-time strategies such as retrieval-augmented generation (RAG) and self-correction. We affirm that the multilayer, defense-in-depth framework incorporating these non-overlapping strategies is crucial for robust hallucination abatement. Some of the ongoing difficulties are the scalable data curation, the trade-off between alignment and model capability, and the problem of editing the reasoning pathways instead of the surface facts.

Keywords: hallucination; large language models (LLMs); factuality; faithfulness; hallucination detection; hallucination mitigation; multimodal AI; AI safety

1. Introduction

The recent advancement in Large Language Models (LLMs) enhance artificial intelligence (AI). As many surveys have covered their development, structure, and uses [1], this trend is now an important step in the evolution from generative to innovative systems [2]. LLM advancement includes novel applications in specialized fields like education, where AI tools are now used to generate instructional video content for complex topics [3]. Despite their capabilities, LLMs can often suffer from hallucinations, meaning that the responses seem reasonable but do not keep to the facts, misrepresent given context, or lack logical consistency [4]. These models are integrated into more critical domains, from information retrieval and medical diagnostics to the complex engineering systems of smart grid communication [5]. Therefore, there has emerged the need for an accurate framework regarding how hallucinations should be understood and classified. This survey begins by reviewing the definitions and taxonomies that have been proposed in the academic literature. Then, it traces the conceptual evolution away from a general notion of factual deviation to a more refined one that distinguishes a

 doi:10.20944/preprints202510.0540.v2

model's internal coherence from its alignment with external reality and provides a key perspective for developing methodologies for effective detection [6]. This paper seeks to clarify the basic ideas behind research on hallucination. It does this by looking at current frameworks and pointing out future paths for strong evaluation and solutions. This work is especially vital for developing reliable AI. It makes sure that LLMs can be used safely in important NLP and decision-making systems.

Table 1. Overview of Hallucination and Factuality Concepts in LLMs, with examples.

Category	Definition	Example / Note
Core Concepts		
Hallucination	Output inconsistent with model's knowledge (context or training)	Fluent but factually or logically wrong
Factuality	Output matches real-world facts	Can be outdated or missing info
Intrinsic vs. Extrinsic		
Intrinsic	Contradicts input context	Wrong data in source summary
Extrinsic	Unsupported by input or training	Fabricated facts beyond knowledge
Faithfulness vs. Factuality		
Faithfulness	Violates user instructions or context	Ignoring prompt or internal contradictions
Factuality	Conflicts with real-world truth	Made-up or false facts
Domain-Specific Types		
Contradiction	Fact or context violation	Entity/relation errors, self-contradiction
Fabrication	Made-up entities/events	False citations, image-text mismatches
Logical Fallacies	Flawed reasoning or code	Invalid logic, dead code

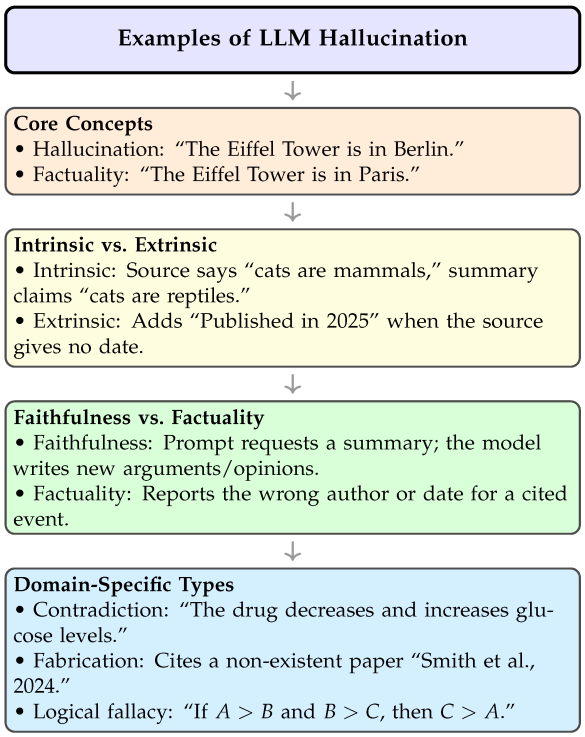

Figure 1. Color-coded conceptual map showing examples of major hallucination and factuality types in LLMs. Shorter arrows indicate hierarchical conceptual flow.

 doi:10.20944/preprints202510.0540.v2

1.1. Definition: From Factual Deviation to Internal Inconsistency

The definition of hallucination describes it as an output from a model that strays from factual reality or includes made-up information. This early understanding worked for the first generation of language models when the main challenge was to stop obvious errors or blatant lies. At that time, hallucinations were mostly seen as simple factual errors. However, today hallucinations have evolved as models have become more coherent and aware of context. They can create content that appears plausible but actually twists logic, data, or the intent of sources. This shift reveals the ongoing importance of the original issues and introduces new challenges that require better solutions at the data, model, and inference levels. As large language models have improved, the original definition has turned out to be too broad, mixing up two types of model failure: failure to be consistent with its available knowledge (hallucination) and failure to be accurate about the world (factuality) [7].

Recent research aims to clarify these two ideas with a more precise classification. Bang et al. specifically define hallucination as output that contradicts the model's available knowledge, training data, and context of inference. In contrast, factuality refers to consistency with verifiable facts in the real world that lie outside the model [8]. This distinction goes beyond mere wording; it influences how we should detect and address hallucinations. A model can generate incorrect output that fits its training data and does not count as a hallucination, or it may accurately generate text that goes against a specific source—representing a true hallucination. Thus, hallucination relates to consistency with known or provided context, while factuality relates to correctness against external reality. Understanding this difference allows for more effective detection: internal consistency checks fundamentally differ from external fact-checking [9].

To provide a high-level map of these concepts, Figure 2 illustrates the conceptual taxonomy of hallucination definitions used in this review. It traces the core definition to its primary analytical frameworks—Intrinsic vs. Extrinsic and Factuality vs. Faithfulness—and the resulting domain-specific types, which are detailed in the following sections.

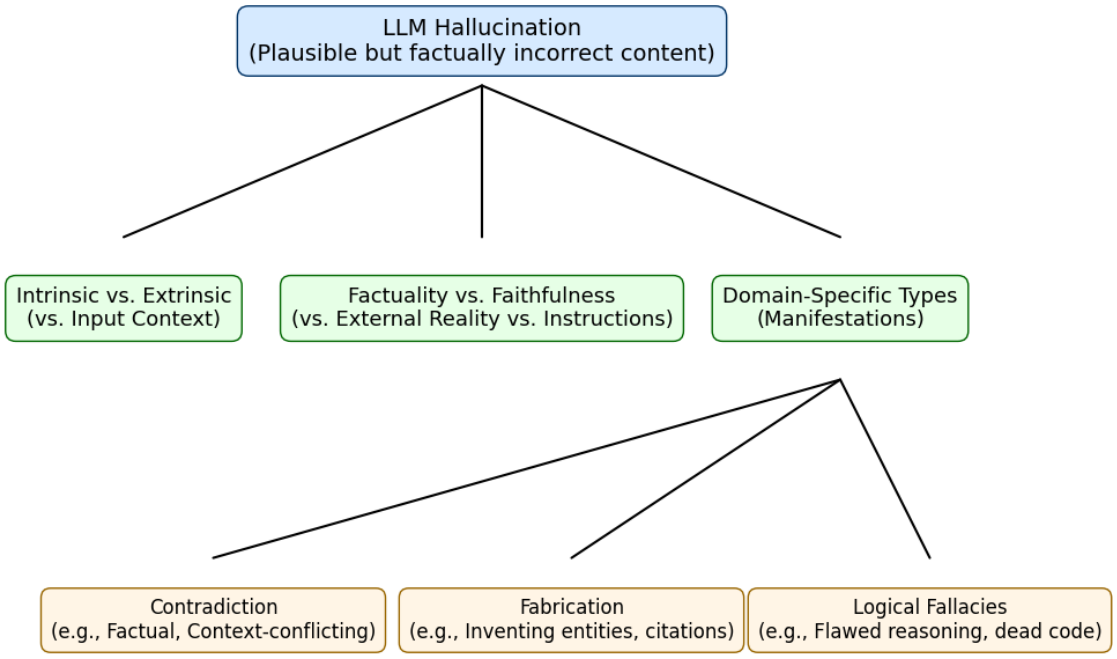

Figure 2. Conceptual Taxonomy of LLM Hallucination. This figure maps the core definition of hallucination to the key analytical frameworks (Intrinsic/Extrinsic, Factuality/Faithfulness) and the specific manifestations (Contradiction, Fabrication, Logical Fallacies).

 doi:10.20944/preprints202510.0540.v2

To illustrate the conceptual difference discussed earlier, Figure 3 places model outputs in a two-dimensional space based on factuality and faithfulness. This framework shows that hallucinations come from factual mistakes and from not following user instructions or maintaining consistency with the given context. On the other hand, a model can provide factually correct information while still being unfaithful to the input. Understanding these dimensions offers a clear way to view future detection and mitigation strategies.

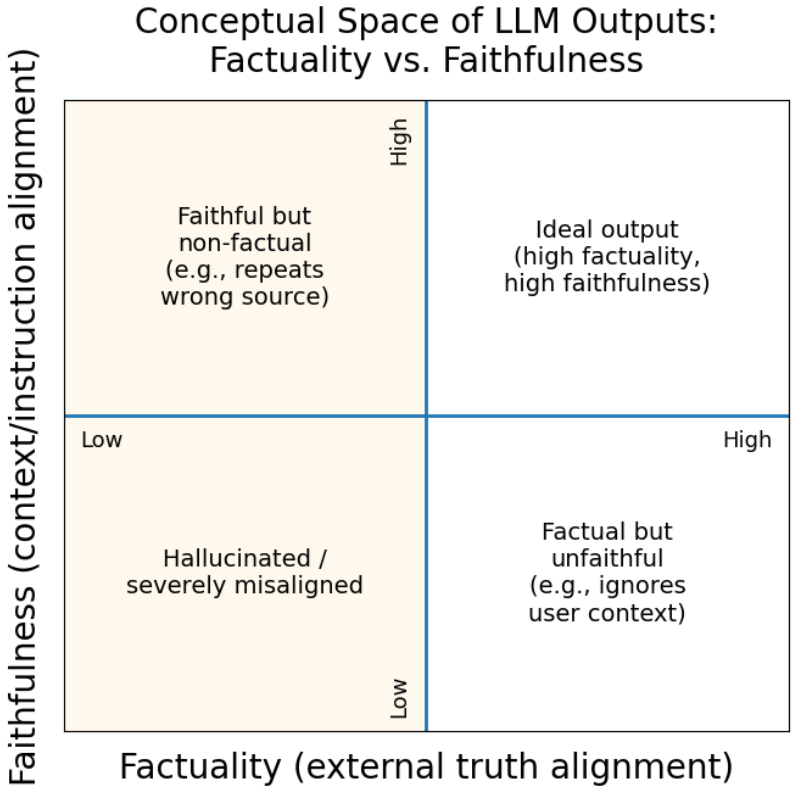

Figure 3. Conceptual space of LLM outputs along factuality and faithfulness. Factuality-horizontal axis-represents alignment with the external world. knowledge, while the vertical axis captures faithfulness to the prompt or provided context. Optimal responses are in the upper-right area, while The lower-left shows outputs either hallucinated or misaligned. The other two quadrants illustrate cases where responses are faithful but wrong, or factual but unfaithful, underlines several reasons why hallucination and factuality must be regarded as different evaluation dimensions.

1.2. Intrinsic vs. Extrinsic

Based on this basic differentiation, two main categories have appeared in the literature for classifying hallucinations: the intrinsic/extrinsic framework and the factuality/faithfulness framework [9].

- **Intrinsic Hallucinations**: An intrinsic hallucination refers to the generation of text that creates content in direct opposition to any input provided by a user. It reflects the failure to be faithful to the immediate context. For example, if the source text states that a company's revenue was $10 million, but the model summary reports it as $15 million, this would be known as an intrinsic hallucination [10]. This category discusses issues with regard to the model's capabilities on the presentation of information available during inference [11].
- **Extrinsic Hallucinations**: This kind of error comes up when the model generates content that could not have been verified from the provided context and is inconsistent with its training data. It involves fabricating unsupported information. For example, in case a summary contains, "The CEO also announced plans to expand into the Asian market," where it was not mentioned in the source, or is very unlikely to be part of training data, it is an extrinsic hallucination. Similar to intrinsic hallucinations, this constitutes another type of error in factuality-whereas now the

 doi:10.20944/preprints202510.0540.v2

error was due to overextrapolation beyond the available information. This underlines the model's inability to realize the boundaries of its inner knowledge [10].

1.3. Factuality vs. Faithfulness Framework

A complementary taxonomy, by Huang et al., [4], classifies hallucinations based on which kind of principle has been violated: faithfulness/adherence to user-provided context or factuality/alignment with objective external facts. Yet, while insightful, this framework can blur the distinction between internal coherence and factual correctness.

Hallucination needs to be differentiated from low factuality, even though these two are generally mutually inclusive. Hallucination is related to whether or not the output of the model makes sense with its accessible knowledge, i.e., the input context or training data, which points out internal incoherence problems. In contrast, factuality concerns the objective validity of the content about external real-world facts [7,12]. A model may be internally coherent in its response but factually wrong due to outdated training data or changes in the real world.

These hallucinations often come out fluent, confident, and persuasive, making detection for both automated mechanisms and human users very complicated. Although some believe that controlled hallucination might be desirable for creative applications, this is not what is desired in high-stakes domains like medicine, law, or finance. In these settings, mitigating hallucinations is key to ensuring the reliability, safety, and ethical deployment of LLMs.

Factuality Hallucination refers to outputs that contradict verifiable real-world knowledge. It includes:

- *Factual inconsistency*: the output directly contradicts known facts [13].
- *Factual fabrication*: the model invents facts not grounded in external sources [4].

Faithfulness Hallucination arises when the generated content diverges from user-provided constraints or the prompt context. Key forms include:

- *Instruction inconsistency*: failure to follow or accurately interpret user instructions.
- *Context inconsistency*: contradiction of the provided input, aligning closely with intrinsic hallucinations.
- *Logical inconsistency*: internal contradictions within the generated output itself.

1.4. Domain-Specific Manifestations of Hallucination

Hallucinations in large language multimodal models are inconsistencies between facts from the visual content and the generated textual output. These hallucinations can manifest as judgment deficiencies, where a generated answer may indicate an incorrect true or false response, or descriptive inaccuracies, as mismatched visual details [14]. A more detailed taxonomy has been considered recently, with LLMs applied to an increasing number of modalities and applications to deal with the various kinds of hallucinations that appear across different domains.

- **Contradiction:** This category captures direct violations of known facts or inconsistencies with provided context [15,16].

  Factual contradiction: When the model generates statements that are contradictory to real-world knowledge [4]. Entity-error hallucinations can include naming an incorrect entity, while relation-error hallucinations can include those that misrepresent the relationship existing between entities.
  Context-conflicting hallucinations: These occur when generated output contradicts either prior model outputs. A common example of this would be when a summary makes an incorrect substitution for a person's name mentioned earlier [17].
  Input-conflicting hallucinations: occur when generated content strays off from what the user has input entirely, by adding details which may not be found in the source material [17].
  context inconsistency: In code generation tasks, hallucinations arise when newly produced code contradicts prior code segments or user instructions, typically manifesting as subtle logical or semantic flaws within the generated program [18,19].

 doi:10.20944/preprints202510.0540.v2

There is a specific subcategory of *self-contradiction* where the model generates two semantically conflicting statements within one response, even though it is conditioned on a unified context [19].

- **Fabrication:** In this type of hallucination, the model generates entities, events, or even citations that are completely fabricated or cannot be verified [20]. This shows that the model tends to create details instead of leaving out or changing the ones that are already there. This clearly breaks the rules of factual accuracy. In the case of multimodal LLMs, fabrication often manifests as mismatches between visual inputs and textual descriptions to create information that was never there in either modality. Two common manifestations are described below:

  Object category hallucinations: The model generates objects not present in the actual scene. In simple terms, one might refer to a "laptop" or "small dog" that is not there. These hallucinations fabricate entirely new visual entities.
  Object attribute hallucinations are when real objects are described with imaginary or exaggerated attributes, such as shape, color, or number, different from reality. These reflect a subtler form of fabrication, where the invented detail replaces an accurate perceptual feature and frequently leads to a misrepresentation of events or a counting error.
  Object relation hallucinations pertain to incorrect assertions of spatial and functional relationships, including and not limited to, "the dog is under the table," when it is beside it [14].
  In medical applications of high stakes where AI is increasingly being used for diagnosis, treatment, and patient monitoring [21–23], fabrication can result in hallucinated clinical guidelines, procedures, or sources that do not exist [24].
  More generally, fabrication occurs when LLMs are prompted with questions beyond their training knowledge or when data is sparse to support certain questions, and they generate content that might sound plausible without supporting this content. For public places, if one broadcasts a false report of fire or any type of calamity with malicious intent, then a state of panic may ensue in the people present there.

- **Logical Fallacies:** These hallucinations reflect flawed reasoning or illogical outputs in tasks requiring step-by-step reasoning [25,26].

  In medicine, these errors fall under the umbrella of *Incomplete Chains of Reasoning*, encompassing:

  * *Reasoning hallucination*—incorrect logic in clinical explanations;
  * *Decision-making hallucination*—unsound treatment suggestions;
  * *Diagnostic hallucination*—medically invalid diagnoses [24].

  In general LLMs, such flaws are often labeled as *logical inconsistencies*, which include internal contradictions within the same output.
  In code generation, this includes:

  * *Intention conflicts*, where the generated code deviates from the intended functionality—either in overall or local semantics [27];
  * *Dead code*, referring to generated segments that are syntactically valid but functionally redundant—statements or blocks that execute no meaningful operation and may cause logical or runtime errors [19,28].

  In multimodal contexts, logical fallacies may appear as judgment errors, such as incorrect answers to visual reasoning tasks.

2. Hallucination Detection Methods

Hallucination detection is a key component in enhancing the factual reliability of large language models. As summerized in Table 2, the detection methods can be grouped according to how directly they interact with the model itself—whether they have access to internal states and gradients (white-box), have limited access to intermediate outputs (grey-box), or treat the model as a completely black-box [15]. Each provides different strengths in interpretability, accuracy, and feasibility of use during real-world deployment.

 doi:10.20944/preprints202510.0540.v2

Table 2. Comparison of Hallucination Detection Methods by Access Level.

Category	Detection Paradigm	Core Principle	Key Methods / Papers	Strengths	Limitations
White-box	Internal State Analysis	Hidden-state activations reveal hallucination-specific patterns	INSIDE (EigenScore), SAPLMA, OPERA, DoLa	Mechanistic signal	Requires full model access; model-specific
Grey-box	Uncertainty Quantification	Low confidence or high entropy indicates hallucination	Token/sequence probability, semantic entropy, LLM-Check	Efficient; logits accessible	Probabilistic assumptions may fail; requires API logits
Black-box	Consistency Checking	Hallucinations vary across outputs or agent disagreement	SelfCheckGPT, LM-vs-LM, multi-agent debate	Model-agnostic	Expensive; fails under consistent hallucination

2.1. Uncertainty-Based Detection

Current research shows a clear shift toward more efficient and targeted solutions. These methods assess whether hallucinations are likely based on the model's uncertainty or inconsistency during output generation.

- **Logit-Based Estimation (Grey-box):** Measures output entropy or minimum token probability. Higher entropy is often correlated with hallucinations [29]. This approach operates on the principle that the model's confidence is encoded in its output probability distribution. Techniques in this category analyze the token-level logits provided by the model's final layer. A high Shannon entropy across the vocabulary for a given token position indicates that the model is uncertain and distributing probability mass widely, which is a strong signal of potential hallucination. Other metrics include using the normalized probability of the generated token itself; a low probability suggests the model found the chosen token unlikely, even if it was selected during sampling.
- **Verbalized Confidence (Black-box):** Prompts the model to self-report confidence levels in its own outputs (e.g., on a 0–100 scale). Useful but sometimes misleading [30,31].
- **Consistency-Based Estimation (Black-box):** Generates multiple completions for the same prompt and computes their agreement via metrics such as BERTScore or n-gram overlap. Used in SelfCheckGPT [32] and variants. For example, SelfCheckGPT generates multiple responses and compares them to the original sentence using metrics like n-gram overlap, BERTScore, or even a question-answering framework to see if questions derived from one response can be answered by others. Anything that is not consistently supported through generations is flagged as a possible hallucination. However, the major shortcoming with this technique is the computational cost, since a single detection requires several inference calls and might also fail if the model keeps reproducing the same factual error in all outputs.
- **Pseudo-Entropy (Grey-box):** Estimates token-level entropy from top-k probabilities returned by APIs, effective in restricted-access settings [33,34].

2.2. Knowledge-Based Detection (Fact-Checking and Grounding)

This class of methods verifies outputs against internal or external factual resources.

- **External Retrieval (RAG-based):** This approach transforms hallucination detection into a fact-checking task by grounding the LLM's output in external, authoritative knowledge. The generated text is first decomposed into verifiable claims, each used to query a retriever that fetches relevant evidence from a corpus (e.g., Wikipedia or a domain-specific database). A verifier—typically an NLI model or an LLM—then determines whether the evidence supports, refutes, or is neutral

 doi:10.20944/preprints202510.0540.v2

toward each claim. Although effective, performance depends heavily on retrieval quality and the reliability of the external knowledge base [35].

- **Internal Verification (Chain-of-Verification):** This black-box method asks the model to review and improve its own output. The steps are: (1) generate an initial answer, (2) create verification questions focusing on its own claims, (3) answer those questions independently, and (4) revise the original response based on these checks. This process encourages self-reflection and helps the model fix inconsistencies. However, its effectiveness is limited by the model's internal knowledge and its ability to create meaningful verification questions [36].
- **Knowledge Graph Validation:** This method checks facts generated by the model against structured knowledge graphs. It is especially useful for verifying relational consistency [37,38].
- **ChainPoll Adherence (Closed-domain):** This approach assesses how well the output matches the provided evidence passages. It uses multi-round chain-of-thought prompting and majority voting [33].

2.3. Dedicated Detection Models

These methods involve training additional models specialized for hallucination detection.

- **QA-Based Fact Checking:** This approach turns model outputs into questions and uses QA pipelines to get answers from the source text. Any differences between the retrieved answers and the model's output suggest possible hallucinations [39].
- **LLM-as-a-Judge:** Using prompts, a strong LLM (e.g., GPT-4) can judge how correct another LLM's output is by using Chain-of-Thought reasoning. ChainPoll Correctness and G-Eval are instances of this [40,41].
- **Supervised Classifiers:** Train models directly on hallucinated vs. non-hallucinated examples [42] (e.g., HALOCHECK [43]).

Making AI easier to understand is getting more and more important, like with those AutoML systems. Some methods use human-readable explanations, such as CoT in ChainPoll or proof in NLI. Looking at Table 3, newer ways to spot problems focus on how the AI thinks (like checking RAG and making sure it's consistent). They don't focus so much on the data itself. This switch means it's cheaper to train and easier to use in different situations because it's quick and done when you need it.

Table 3. Representative Techniques for Detecting Hallucinations in LLMs.

Technique	Detection Type	Reference
SelfCheckGPT	Consistency-based (Black-box)	[44]
ChainPoll	Prompt-based Judge (Black-box)	[33]
G-Eval	Prompted LLM with scoring aggregation	[45]
RAG + Verifier	Retrieval-based Fact-checking (Grey/Black-box)	[46]
Chain-of-Verification	Internal self-questioning (Black-box)	[36]
Graph-based Context-Aware (GCA)	Reference alignment via graph matching	[47]
ReDeEP	Mechanistic interpretability in RAG	[48]
Drowzee (Metamorphic Testing)	Logic-programming and metamorphic prompts	[49]
MIND (Internal-state monitoring)	Internal activations	[50]
Verify-when-Uncertain	Combined self- and cross-model consistency (Black-box)	[31]
Holistic Multimodal LLM Detection	Bottom-up detection in Multimodal LLMs	[51]
Large Vision Language Models Hallucination Benchmarks/M-HalDetect	Multimodal reference-free detection	[52]

 doi:10.20944/preprints202510.0540.v2

2.4. Summary and Practical Considerations

Figure 4 illustrates that selecting an appropriate hallucination detection method requires balancing trade-offs between cost, access, and explainability. Key factors guiding this choice include:

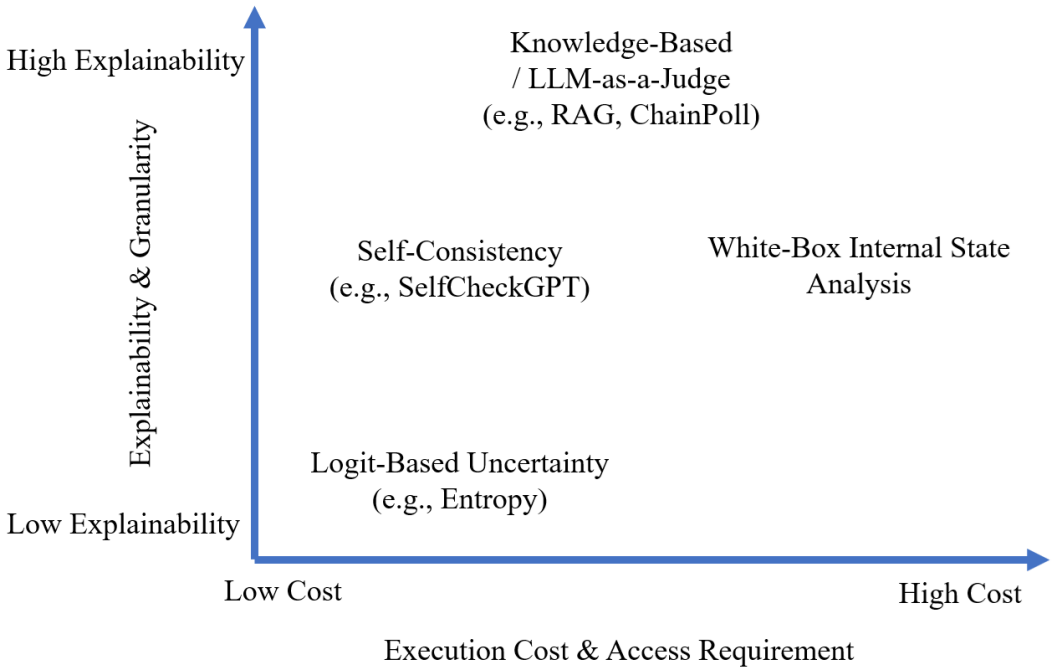

Figure 4. A conceptual landscape of hallucination detection methods, mapping techniques based on their trade-offs between execution cost/access requirements and the explainability of their outputs. Methods in the top-left are highly explainable and accessible via standard APIs, while methods on the right require privileged model access or high computational cost.

- **Application Scope:** Whether the hallucination is open-domain (factual world knowledge) or closed-domain (reference-consistent).
- **Model Access:** Varying from full access to weights (white-box) to restricted API-only access (black-box).
- **Efficiency:** Cost of multiple generations (e.g., 20 runs in SelfCheckGPT vs. 5 in ChainPoll).
- **Explainability:** This refers to whether human-readable justifications are provided (e.g., CoT in ChainPoll or entailment evidence in NLI). This emphasis on transparency aligns with the growing demand for user-friendly, explainable frameworks in AI-driven processes, such as those emerging in AutoML [53].

As LLMs are widely used, hallucination detection methods must be scalable, reliable across diverse tasks, and adaptable to new model architectures.

Table 4 summarizes the dominant categories of hallucination sources, the corresponding detection strategies, and mitigation methods. This table provides a unified view of how detection and mitigation techniques connect to the underlying causes of hallucination. This mapping highlights how different parts of the LLM pipeline (e.g., data, model, and inference) require complementary solutions. The table also provides key references across the literature to guide readers toward influential work in each category.

 doi:10.20944/preprints202510.0540.v2

Table 4. Summary of Hallucination Sources, Detection Strategies, and Mitigation Techniques in LLMs.

Hallucination Source	Detection Strategy	Mitigation Technique	Representative Works
Data Issues (low-quality, noisy, or missing evidence)	Grey-box uncertainty (entropy, token probabilities); Black-box consistency (SelfCheckGPT); Evidence retrieval mismatch	Data filtering, deduplication; Source up-weighting; Retrieval-augmented pretraining (RETRO)	[4,54,55]
Model Limitations (parametric knowledge errors; insufficient grounding)	Internal activation-based detection (white-box); Model probing (EigenScore, OPERA)	Supervised fine-tuning on fact-checked data; Preference optimization (RLHF, DPO); Knowledge editing (ROME, MEMIT, IFMET)	[56–58]
Inference-Time Drift (decoding errors; over-sampling; loss of context)	Pseudo-entropy; Self-verification (CoVe); Cross-model agreement (LM-vs-LM)	Better decoding (DoLa, CAD, ITI); Self-correction (Self-Refine); Inference-time retrieval (RAG)	[36,59,60]
Open-Domain Factuality Gaps	RAG-based claim verification; NLI-based fact checking; KG-consistency analysis	RAG + verifier pipelines; Graph-based validation; QA-based fact-checking	[39,46,47]
Closed-Domain Grounding Errors	Evidence-alignment scoring (ChainPoll); Truthfulness consistency; Document adherence tests	Document-grounded CoT; Multi-step verification; Context-adherence penalties	[31,33,34]

3. Hallucination Mitigation Strategies

Following the identification and detection of hallucinations, the next important challenge is to reduce them. Creating believable yet factually incorrect or baseless content remains one of the biggest hurdles to the safe and broad use of LLMs. Reducing this issue is vital in high-stakes areas such as medicine, finance, and law [61]. It is not a one-time fix but a complex process that must be handled throughout the model lifecycle. Effective strategies are necessary to ensure reliability, safety, and ethical use.

Arranging these reduction strategies by lifecycle stage—data-centric, model-centric, and inference-time—offers a clear view of where changes can be made and what trade-offs may arise. This approach shows how each stage brings different costs, limitations, and chances to enhance factual reliability, making it the most useful framework for understanding and applying hallucination reduction in LLMs.

The following techniques for reducing hallucinations are organized by the stage of the model lifecycle at which they are applied: (1) Data-Centric and Pre-Training Strategies , which improve the quality and factual basis of the data used to train LLMs; (2) Model-Centric Strategies, which adjust model parameters through fine-tuning and alignment to encourage truthful behavior; and (3) Inference-Time Strategies , which add post hoc methods during generation to lessen hallucinations [4]. These categories are not separate; instead, strong solutions often combine methods from different stages to create a layered defense.

Table 5 summarizes this taxonomy, highlighting key principles, representative techniques, and associated trade-offs. As the table shows, recent research increasingly emphasizes inference-time mitigation—approaches such as RAG and ITI—because they offer practical, low-cost adaptability compared to data- and model-centric strategies. This trend reflects a broader shift toward modular,

 doi:10.20944/preprints202510.0540.v2

post-training control, where hallucination is managed dynamically during generation rather than being fully eliminated at training time.

Table 5. Taxonomy of Hallucination Mitigation Techniques.

Lifecycle Stage	Core Principle	Representative Techniques	Strengths	Limitations
Data-Centric (Pre-Training)	Improve factual quality of training data to reduce hallucination at the source.	High-quality data curation; retrieval-augmented pre-training (e.g., RETRO [62]).	Reduces hallucination fundamentally; better internal grounding.	High computational cost; limited by source quality; difficult to scale.
Model-Centric (Fine-Tuning & Alignment)	Align model parameters with factual knowledge and human preferences.	Supervised fine-tuning (SFT); RLHF; DPO; knowledge editing (ROME [56,63], MEMIT [57]).	Enables steerability and surgical correction; enhances safety.	Requires human preference data; risk of forgetting; unstable optimization.
Inference-Time	Guide generation with external evidence or real-time reasoning.	Retrieval-augmented generation (RAG); self-refinement (CoVe [36], Self-Refine [64]); decoding control (DoLa [59], CAD [65]); ITI [66].	Model-agnostic; low deployment barrier; can use live information.	Latency overhead; tool quality bottleneck; limited correction for internal errors.

3.1. Data-Centric and Pre-Training Strategies

The most fundamental way to mitigate hallucinations is to address them at their root: the training data [67]. An LLM's propensity to produce non-factual content often reflects the quality of the data it has learned from. Models trained on biased, low-quality, or inaccurate data are inherently more prone to hallucination. Improving the factual integrity of training corpora reduces the need for complex downstream interventions.

3.1.1. High-Quality Data Curation and Filtering

The factual accuracy of LLMs is tightly linked to the quality of their training data. Manual filtering of massive datasets, often trillions of tokens, is infeasible, making automated and heuristic-based strategies critical for large-scale curation.

Common approaches include filtering web-scale corpora (e.g., Common Crawl) to retain only documents from trusted sources such as Wikipedia, academic publications, or reputable books, as well as upsampling verified domains to increase the proportion of reliable content [68].

In addition to filtering, synthetic data generation has become increasingly popular. For example, models like `phi-1.5` are trained on textbook-style synthetic data that includes commonsense reasoning and factual content [69,70]. Data augmentation techniques, especially those that ensure topic diversity and structural coherence, are also important during fine-tuning.

In high-stakes fields like healthcare and law, expert-curated, domain-specific data sets (e.g., MedCPT [24], verified legal documents) are essential. Strict data-governance practices, including dataset versioning and prompt logging, also improve reliability. However, manual or rule-based filtering at this scale is prohibitively expensive and hard to maintain for modern LLM pipelines. These challenges have pushed researchers toward more automated and flexible data-curation strategies, which aim to maintain factual accuracy while improving scalability and efficiency.

 doi:10.20944/preprints202510.0540.v2

3.1.2. Retrieval-Augmented Pre-Training

A better approach includes retrieval in the pre-training process itself. This method allows models to base generation on reliable external sources from the beginning.

A well-known example is the Retrieval-Enhanced Transformer (RETRO). It adds a retrieval module to a large language model (LLM) that accesses a vast collection of supporting documents during training. By integrating retrieved evidence directly into the learning process, RETRO achieves greater factual accuracy than similarly sized models that do not use retrieval. However, this comes with higher demands for computation and storage.

This strategy is resource-heavy, increasing training costs by up to 25%. It also has limitations based on the retrieval sources. If the external memory includes outdated or incorrect information, these mistakes can be incorporated into the model's parameters [55].

The principle of "garbage in, garbage out" is particularly relevant for large language models. Models trained on flawed datasets need strategies during inference to correct built-in inaccuracies, which adds latency and complexity. High-quality data is essential for effectively reducing hallucinations.

3.2. Model-Centric Strategies: Fine-Tuning and Alignment

Model-centric strategies change parameters after pre-training to promote fact-based behaviors. They serve as a compromise; they are more flexible than strict data curation but more reliable than fixes that rely only on inference. We concentrate on three approaches: supervised fine-tuning, preference optimization, and knowledge editing.

3.2.1. Supervised Fine-Tuning (SFT)

SFT adapts a general-purpose model to specific tasks or domains using curated (prompt, response) datasets. Factual accuracy depends heavily on the dataset's quality.

Several techniques enhance SFT's effectiveness for factuality: fine-tuning on fact-checked data from encyclopedias or scientific sources; knowledge injection using stronger teacher models to distill information into weaker students; and training with counterfactuals to improve truth discrimination [71].

Instruction fine-tuning is another promising direction, teaching models to follow structured prompts (e.g., requiring citations or disclaimers when uncertain). However, SFT can introduce challenges such as *catastrophic forgetting*, where new knowledge overwrites useful prior capabilities.

3.2.2. Preference Optimization for Alignment

Rather than training on labeled data, preference optimization aligns models with human judgments of what constitutes better, more factual outputs.

Reinforcement Learning from Human Feedback (RLHF) is the dominant approach, but can inadvertently reward fluency and confidence over accuracy. Variants like Reinforcement Learning for Hallucination (RLFH) decompose outputs into atomic facts, evaluating each for correctness to provide token-level rewards or penalties.

To address this, more targeted methods like **Reinforcement Learning for Hallucination (RLFH)** decompose outputs into atomic facts, evaluate each for correctness, and propagate token-level rewards or penalties accordingly. These methods illustrate the trade-off between targeted factual correction and generalization capacity—a tension that defines current alignment research.

Direct Preference Optimization (DPO) has emerged as a simpler and more stable alternative to RLHF. It reframes reward learning as a classification task over preference pairs (e.g., factual vs. hallucinated responses), avoiding the need for a reward model or RL algorithm. DPO achieves comparable or superior performance and has inspired variants such as: **Cal-DPO**: calibrates implicit reward scales for more controlled updates [72]. **V-DPO**: incorporates visual context to reduce hallucinations in vision-language models [73].

These methods highlight the increasing sophistication of model alignment techniques and their importance in reducing errors in deployed systems.

 doi:10.20944/preprints202510.0540.v2

The rapid move from RLHF to DPO shows a major trend in LLM alignment: the change from complex, multi-step, and often unstable processes to simpler, more efficient options. RLHF usually involves several steps—supervised fine-tuning, collecting preference data, training a reward model, and reinforcement learning. Each step brings its own challenges in implementation and sensitivity to settings.

DPO's main innovation is removing the intermediate reward modeling step by redefining alignment as a direct optimization over preference pairs, similar to a classification goal. This simplification maintains or even improves performance, suggesting that the advantages of complex, bio-inspired thinking can often be captured through more straightforward math. The broader implication is a shift towards scalability and practical engineering in alignment research.

This trend points to a future where alignment goals may be included earlier in the training process—possibly through preference-aware pre-training—reducing the need for expensive after-the-fact fixes and integrating alignment more naturally into model development.

3.2.3. Knowledge Editing: Surgical Model Updates

Knowledge editing provides a targeted approach to fine-tuning. It allows for specific changes to an LLM's internal knowledge without the need for retraining or the risk of losing previous information. These methods focus on updating or correcting individual facts, like changing a CEO's name or a capital city, while keeping the rest of the model's behavior intact.

Most techniques follow a locate-then-edit approach [74]. By using mechanistic interpretability tools, such as causal tracing, these methods find the specific MLP layers or neurons that hold a factual association. Once they identify these areas, they directly change the weights to reflect the new information. Prominent examples include:

- **ROME (Rank-One Model Editing):** Introduces a rank-one update to a single MLP layer to revise a single fact [56].
- **MEMIT (Mass Editing Memory in Transformers):** Extends ROME to modify multiple layers, allowing batch editing of thousands of facts [57].
- **GRACE:** Stores updated knowledge in an external memory module without modifying model parameters [75].
- **In-Context Knowledge Editing (IKE):** A training-free approach that injects new facts into the prompt context during inference [76].

While these techniques differ in how they work, they all show a shift toward modular and interpretable control over model behavior. Although they are effective for simple factual updates, locate-then-edit methods struggle with multi-hop reasoning. Research indicates that shallow layers often store basic facts, while deeper layers manage reasoning chains and indirect conclusions. As a result, editing a fact from a shallow layer (for example, "Paris is the capital of France") may not update the deeper reasoning pathways required to answer questions like "What is the capital of the country where the Eiffel Tower is located?"

Newer methods like **IFMET** [74] (Interpretability-based Factual Multi-hop Editing in Transformers) aim to tackle this issue by identifying and updating both shallow and deep layers. This approach improves performance on multi-step reasoning tasks. However, no single editing method currently excels in all important areas: effectiveness, generalization, location, and performance. The model architecture and domain still rely heavily on the specific model being used.

The locate-then-edit approach serves as a strong but limited metaphor. It treats knowledge in LLMs as editable "code." Techniques like ROME and MEMIT offer evidence for this perspective, as they show that factual connections are often localized and can be edited. Nonetheless, their limitations in multihop reasoning [77] highlight the metaphor's shortcomings. Updating one "line of code" (for example, "A is the parent of B") does not recompile the entire "program" to reflect downstream conclusions (like "A is the grandparent of C").

This indicates that storing facts and reasoning through facts are separate yet intertwined processes within transformer models. Shifting from "fact editing" to *reasoning path editing* [78] is a promising

 doi:10.20944/preprints202510.0540.v2

area for future research. This shift will require better interpretability tools that can trace and modify complete causal chains of computation within the network.

3.3. Inference-Time Mitigation Strategies

Inference-time mitigation strategies operate during text generation and offer a practical advantage: they do not require model retraining or parameter updates. These methods are often model-agnostic and applicable to proprietary or black-box APIs, making them attractive for deployment in real-world systems.

Broadly, these techniques fall into three categories:

1. **Knowledge-Augmented Generation:** Augment the prompt with retrieved external evidence.
2. **Structured Reasoning and Self-Correction:** Guide the generation process through explicit, multi-step reasoning [79].
3. **Advanced Decoding and Intervention:** Modify the decoding process or directly intervene in the model's internal activations [80].

3.3.1. Retrieval-Augmented Generation (RAG)

RAG grounds model responses in external, verifiable sources. It follows a *retrieve-augment-generate* pipeline:

1. Retrieve: Search a knowledge base (e.g., via dense retrieval) using the user query.
2. Augment: Append retrieved evidence to the prompt.
3. Generate: Produce a response conditioned on both the query and the retrieved documents.

Modern RAG systems go beyond this basic structure. For instance:

- **Iterative RAG:** Performs multiple rounds of retrieval and generation.
- **Self-Corrective RAG:** Produces an initial draft followed by evidence retrieval for iterative self-revision.

While powerful, Retrieval-Augmented Generation (RAG) introduces new failure points, collectively termed RAG-induced hallucinations, which can manifest at various stages of the pipeline. Oftentimes, the process first breaks down at the retrieval stage, where the retriever may fail to fetch relevant documents due to a semantic mismatch with the user's query (low recall), or it might retrieve noisy, irrelevant, or even contradictory documents (low precision). When such flawed context is passed to the generator, it can distract the model, leading to responses that are ungrounded or based on the wrong information. Furthermore, generation failure can occur even with perfectly relevant documents. The LLM might disregard the provided context and revert to its own parametric knowledge, misinterpret the retrieved text leading to a logically inconsistent response, or "over-extrapolate" from the evidence by fabricating plausible details that are not explicitly supported, a common issue in long-form generation tasks. Ultimately, the integrity of the entire RAG system depends on its knowledge base. If the source documents contain factual errors, biases, or outdated information, the RAG system will faithfully reproduce this misinformation under a veneer of authority. A significant challenge also arises when retrieved documents present conflicting information, forcing the model to synthesize an answer from contradictory sources, which can itself lead to hallucinations.

3.3.2. Structured Reasoning and Self-Correction

These techniques impose structure on the model's generation to improve factuality and logical coherence.

Chain-of-Thought (CoT) prompting encourages step-by-step reasoning before providing an answer. It is particularly effective for large models on tasks requiring arithmetic or commonsense reasoning. However, it can degrade performance in smaller models and sometimes masks internal errors, complicating hallucination detection.

 doi:10.20944/preprints202510.0540.v2

Chain-of-Verification (CoVe) introduces an explicit verification loop:

1. Generate a baseline answer.
2. Plan verification questions.
3. Independently answer the questions.
4. Revise the original response based on verification.

This approach encourages the model to fact-check itself and revise its outputs based on new evidence.

CoVe signifies a structured, multi-stage self-verification pipeline that promotes a model to think critically about its initial response [36]. Instead of relying on a single forward pass, CoVe explicitly decomposes the generation process into four phases: (1) generating an initial baseline answer, (2) crafting verification questions targeting that answer, (3) independently answering those questions to identify inconsistencies, and (4) revising the initial response based on the evidence so derived. This feedback loop converts the LLM from a single-shot generator to a self-auditing reasoning machine, greatly mitigating hallucinations in tasks that demand factual precision and multi-hop consistency. Figure 5 summarizes the workflow.

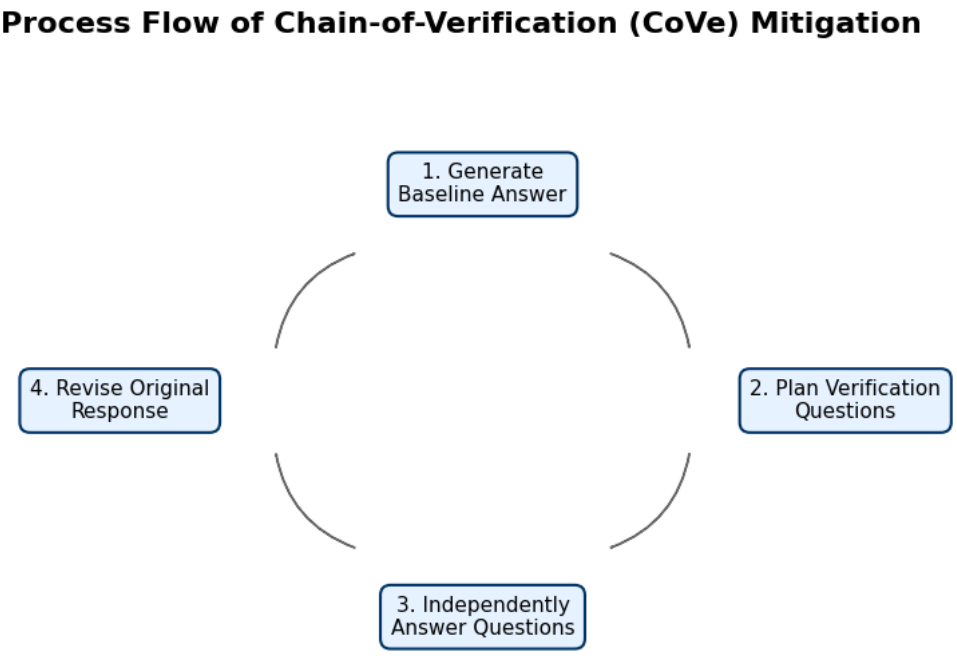

Figure 5. Process flow of the Chain-of-Verification mitigation strategy. The work cycle is divided into four stages: (1) generating an initial baseline answer, (2) planning verification questions to challenge the response, (3) independently answering each verification question, -identifying information that is missing and required for the completeness of the message, -selecting words carrying appropriate meanings relevant to the intended message, -producing expressions grammatically correct, and editing the original response according to verification evidence. This structured self-auditing loop improves response factuality and internal consistency.

Self-Refine generalizes this idea with an iterative framework: *Generate* → *Feedback* → *Refine*. The model plays both author and critic, progressively improving its outputs [64].

These methods represent a shift toward treating the LLM as a programmable reasoning engine. Rather than issuing a single query and receiving a flat response, developers orchestrate multistage workflows with explicit roles, planning, verification, and correction at each stage. This "LLM-as-CPU" paradigm enables more transparent, controllable, and reliable AI systems.

 doi:10.20944/preprints202510.0540.v2

3.3.3. Advanced Decoding and Intervention Strategies

These techniques intervene directly in the model's output probabilities or internal representations, typically requiring logit or layer-level access.

- **DoLa (Decoding by Contrasting Layers):** Amplifies factual content by contrasting logit distributions from mature (later) and premature (earlier) layers, suppressing shallow patterns and emphasizing deep semantic knowledge [59].
- **CAD (Context-Aware Decoding):** Forces the model to attend to provided evidence by penalizing tokens that would have been generated in the absence of context. This encourages grounding in retrieved or injected information, reducing reliance on parametric priors [65].

These decoding-aware methods enable finer-grained control over generation and offer promising directions for mitigating hallucinations, particularly when paired with external retrieval or structured prompting.

A second, more invasive class of inference-time methods involves direct intervention in the model's internal activations.

- **Inference-Time Intervention (ITI):** This white-box technique actively steers the model's internal state toward more truthful representations. ITI applies linear probing to a dataset such as TruthfulQA [81] to identify a sparse set of attention heads whose activations correlate strongly with truthful responses. During inference, it adds a small, learned vector to the output of these attention heads at each generation step. This vector gently nudges the model's activations toward truth-aligned directions, enhancing factual consistency with minimal computational overhead and without requiring any parameter updates.

3.4. A Unified View and Future Directions

The wide range of hallucination mitigation strategies—from data-centric techniques and model editing to inference-time interventions—illustrates that there is no universal solution. Instead, the most robust systems will likely emerge from a *defense-in-depth* approach, where complementary techniques are layered across the model development and deployment pipeline to address different failure modes.

A conceptual model of such a multi-layered mitigation stack includes:

1. **Foundation Layer (Data):** High-quality pre-training data serves as the first line of defense. This layer emphasizes automated filtering, deduplication, and upsampling of trusted sources to ensure a strong factual grounding in the parametric knowledge of the model.
2. **Core Layer (Model Alignment):** Alignment methods like Direct Preference Optimization (DPO) shape the model's behavior toward factual and safe outputs. For known factual inaccuracies, targeted *Knowledge Editing* methods (e.g., ROME, MEMIT) provide localized corrections without retraining the entire model.
3. **Application Layer (Inference-Time Grounding):** Retrieval-Augmented Generation (RAG) and related techniques are applied at deployment to supplement parametric knowledge of the model with up-to-date domain-specific information, helping mitigate knowledge gaps and ensure temporal relevance.
4. **Guardrail Layer (Verification and Post-processing):** Structured reasoning strategies like Chain-of-Verification (CoVe) enforce internal consistency through self-checking. Finally, detection tools serve as a post-hoc safety net, flagging hallucinated outputs for human review or triggering fallback behaviors (e.g., "I don't have enough information to answer").

Despite significant progress, several open challenges and promising research directions remain:

- **Scalable Data Curation:** Developing fully automated, scalable, and multilingual pipelines for high-quality data curation remains a foundational challenge, especially at trillion-token scales.
- **The Alignment–Capability Trade-off:** Overalignment may suppress the model's general capabilities—such as reasoning or creativity—resulting in an "alignment tax." Research is needed to develop alignment strategies that maintain or even enhance model utility across tasks.

 doi:10.20944/preprints202510.0540.v2

- **Editing Reasoning, Not Just Facts:** Most knowledge editing methods focus on changing discrete facts. However, achieving true robustness requires tracing how those facts shape a model's reasoning and directly modifying the pathways that drive its behavior—a task that calls for deeper mechanistic interpretability.
- **Compositionality of Mitigation Techniques:** While combining mitigation methods appears beneficial, their interactions are poorly understood. For example, how does RAG interact with ITI or DPO? Can editing methods interfere with self-correction routines? Systematic analysis is needed to develop principled strategies for composing techniques effectively.
- **The Inevitability of Hallucination:** An emerging perspective suggests that hallucination may be an inherent artifact of probabilistic next-token generation and the current training paradigm. If so, mitigation efforts may shift from eliminating hallucinations to managing them through uncertainty estimation, robust detection, fallback policies, and human-in-the-loop oversight. This view challenges the assumption that hallucination can ever be fully eradicated and instead reframes it as a fundamental trade-off between creativity and factual reliability—an insight that has profound implications for the design of future alignment and interpretability research.

To consolidate the preceding sections, Figure 6 provides an integrated multi-layer mitigation stack that aligns with the model life cycle. This visualization reflects the defense-in-depth perspective that was introduced in Section 3, highlighting how data-centric, model-centric, and inference-time interventions interact to form a coherent approach to reducing hallucination. The figure situates each mitigation technique in its respective lifecycle layer and provides a Practical road map for researchers and practitioners on the choice of interventions according to their deployments. constraints, available resources, and demands for fact reliability.

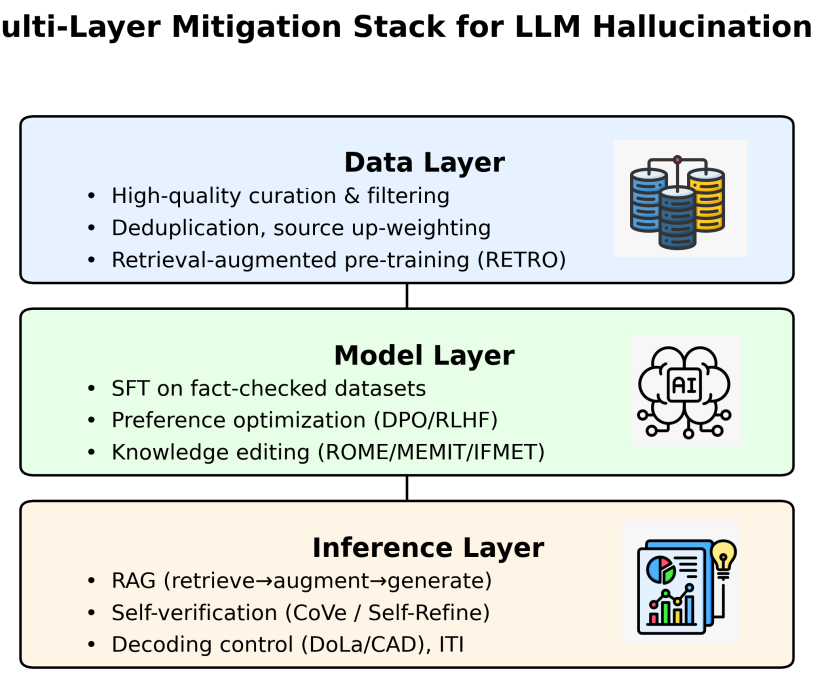

Figure 6. Multi-Layer Mitigation Stack for LLM Hallucination. The figure illustrates the three major intervention layers—Data, Model, and Inference—and examples of representative techniques. At the *Data Layer*, high-quality curation, deduplication, and retrieval-augmented pre-training form the foundation for reducing hallucination at its source. The *Model Layer* adds alignment and parametric updates such as SFT, preference optimization (e.g., DPO/RLHF), and knowledge-editing methods (ROME, MEMIT, IFMET). Finally, the *Inference Layer* provides real-time grounding and control through RAG, self-verification, and decoding-intervention techniques (e.g., DoLa, CAD, ITI). These layers taken together provide a defense-in-depth framework for practical hallucination mitigation.

 doi:10.20944/preprints202510.0540.v2

4. Recent Trends and Open Issues

The field of LLM hallucination research is changing quickly. It is transitioning from basic definitions to more developed, connected, and scalable solutions. Although we have made notable progress in spotting and reducing hallucinations, key trends and ongoing problems define the current research landscape. This section highlights the main emerging trends in detection and mitigation, as well as the unresolved challenges that will influence future efforts.

4.1. Emerging Trends in Mitigation and Detection

Current research shows a clear shift toward more efficient, targeted, and system-level ways to manage hallucination. Four major trends are evident:

- **Shift Toward Simpler Alignment Methods:** There is a clear trend away from complex, multi-stage alignment pipelines like Reinforcement Learning from Human Feedback (RLHF). People are moving toward simpler, more stable, and mathematically straightforward alternatives. The rapid adoption of **Direct Preference Optimization (DPO)** and its variants exemplifies this, as it achieves comparable or superior performance to RLHF without the need to train a separate reward model, thus reducing complexity and computational overhead.
- **Rise of Surgical Knowledge Editing:** Rather than depending on expensive full-model fine-tuning, researchers are increasingly creating "surgical" methods to directly edit a model's internal knowledge. Techniques like **ROME** and **MEMIT** use mechanistic interpretability to find and accurately change the model parameters that store specific facts. This enables the efficient correction of factual errors without the risk of losing other knowledge.
- **Advanced Inference-Time Interventions:** A major area of innovation is developing techniques that work during the generation process without needing parameter updates. These methods are very practical because they can apply to any model, including proprietary APIs. This trend includes Retrieval-Augmented Generation (RAG) pipelines that base outputs on external evidence, self-correction routines [82] like Chain-of-Verification (CoVe), which encourage a model to check the accuracy of its own statements, and decoding strategies (e.g., DoLa, ITI) that adjust a model's internal activations to guide it toward more truthful outputs.
- **Development of LLM-as-a-Judge Paradigms:** For detection, there is an increasing tendency to use powerful LLMs as evaluators. The LLM-as-a-Judge [40,83] paradigm, evident in methods like G-Eval [45], utilizes the reasoning abilities of top models (e.g., GPT-4) to evaluate the accuracy and reliability of outputs from other models. This often surpasses traditional metrics and gets close to human-level judgment.

4.2. Key Open Problems

Despite significant progress, several foundational challenges remain in hallucination research:

- **Scalable and High-Quality Data Curation:** The principle of "garbage in, garbage out" continues to be a major hurdle. Creating fully automated, scalable, and multilingual pipelines to gather high-quality, factually correct, and diverse training data is a key challenge that has not been solved, especially when dealing with trillions of tokens.
- **The Alignment-Capability Trade-off:** A significant issue is the risk of an "alignment tax." This happens when efforts to improve accuracy and reduce mistakes unintentionally weaken other important abilities of the model. Strong fine-tuning or preference adjustments, like RLHF or DPO, can make a model too cautious. It may refuse to answer reasonable questions or lose its ability to carry out complex, multi-step reasoning. For example, a model that is heavily focused on accuracy might become less creative or struggle with tasks requiring subtle reasoning beyond what is explicitly stated. This creates a tough challenge: how can we make models more truthful without making them less useful? Future research should concentrate on developing alignment methods that are more precise and less disruptive. This could mean separating different model abilities at a mechanical level, allowing for targeted adjustments that improve accuracy without

 doi:10.20944/preprints202510.0540.v2

damaging reasoning or creativity. Another promising approach is to create alignment methods that encourage nuanced behaviors, such as showing uncertainty or offering conditional answers, instead of just punishing outputs that cannot be verified. Balancing accuracy with usefulness remains a key challenge for the next generation of LLMs.

- **Editing Reasoning Paths, Not Just Facts:** Current knowledge editing techniques, such as STRUEDIT [84], are good at correcting simple, single-step facts, like "Paris is the capital of France." However, they have difficulty updating the intricate, multi-step reasoning paths that rely on those facts. Moving from fact-editing to reasoning-path editing is a significant challenge. This will need better tools for understanding how these systems work.
- **Compositionality of Mitigation Techniques:** The strongest systems will probably use a "defense-in-depth" strategy that includes various mitigation techniques. However, we do not fully understand how these methods interact. Research is needed to gain a solid understanding of how different strategies, such as RAG, DPO, and knowledge editing, can work together effectively without disrupting each other.

5. Conclusions

This review gives a clear look at the complex issue of hallucination in Large Language Models. It covers everything from basic definitions to effective ways to detect and reduce these problems. Our analysis shows an important shift in perspective. We now see hallucinations not just as simple factual mistakes but as failures to stay true to a model's known context, which is different from external facts. We grouped the various detection methods based on their access requirements. This highlighted a key trade-off: the high accuracy of white-box methods that focus on interpretability versus the wider use of black-box methods that rely on consistency.

In exploring ways to reduce hallucinations, we found that there isn't a single answer. The best approach involves a layered strategy called "defense-in-depth." This means using multiple techniques throughout the model's lifecycle. It starts with careful curation of high-quality, fact-based data (data-centric), continues with strong model alignment through preference optimization and targeted knowledge editing (model-centric), and finishes with real-time, evidence-based grounding during deployment (inference-time). Despite making significant strides, we still face major challenges. Future efforts need to focus on scalable data curation, the so-called "alignment tax," where attempts to reduce hallucination may limit the model's capabilities, and the tricky task of not just editing facts but also the reasoning paths that produce them. Ultimately, handling hallucination doesn't mean eliminating it completely. It's about creating a strong, layered system of tools and practices that make LLMs reliable, clear, and safe for broad use in society.

Author Contributions: Conceptualization, S.M.S.M., B.C.K., C.E., and O.K.; methodology, S.M.S.M., B.C.K. and C.E.; writing—original draft preparation, S.M.S.M., B.C.K., C.E. and O.K.; writing—review and editing, S.M.S.M., B.C.K., C.E. and O.K.; visualization, S.M.S.M., B.C.K. and C.E.; supervision, C.E., and O.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC was funded by Cardiff University Institutional Funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created in this study. Data sharing is not applicable to this article.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest. This research has not received any specific grant from public funding agencies or commercial or not-for-profit sectors.

 doi:10.20944/preprints202510.0540.v2

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
CAD	Context-Aware Decoding
Cal-DPO	Calibrated Direct Preference Optimization
CoT	Chain-of-Thought
CoVe	Chain-of-Verification
DPO	Direct Preference Optimization
GCA	Graph-based Context-Aware
ICE	In-Context Editing
IFMET	Interpretability-based Factual Multi-hop Editing in Transformers
ITI	Inference-Time Intervention
KG	Knowledge Graph
LLM	Large Language Model
MEMIT	Mass Editing Memory in Transformers
MIND	Internal-state monitoring
MLLM	Multimodal Large Language Model
MLP	Multi-Layer Perceptron
RAG	Retrieval-Augmented Generation
RETRO	Retrieval-Enhanced Transformer
RLHF	Reinforcement Learning from Human Feedback
RLFH	Reinforcement Learning for Hallucination
ROME	Rank-One Model Editing
SFT	Supervised Fine-Tuning
V-DPO	Vision-guided Direct Preference Optimization

References

1. Sajjadi Mohammadabadi, S.M.; Kara, B.C.; Eyupoglu, C.; Uzay, C.; Tosun, M.S.; Karakuş, O. A Survey of Large Language Models: Evolution, Architectures, Adaptation, Benchmarking, Applications, Challenges, and Societal Implications. *Electronics* **2025**, *14*. https://doi.org/10.3390/electronics14183580.
2. Mohammadabadi, S.M.S. From generative ai to innovative ai: An evolutionary roadmap. *arXiv preprint arXiv:2503.11419* **2025**.
3. Maleki, E.; Chen, L.T.; Vijayakumar, T.M.; Asumah, H.; Tretheway, P.; Liu, L.; Fu, Y.; Chu, P. AI-generated and YouTube Videos on Navigating the US Healthcare Systems: Evaluation and Reflection. *International Journal of Technology in Teaching & Learning* **2024**, *20*.
4. 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 Transactions on Information Systems* **2025**, *43*, 1–55.
5. Mohammadabadi, S.M.S.; Entezami, M.; Moghaddam, A.K.; Orangian, M.; Nejadshamsi, S. Generative artificial intelligence for distributed learning to enhance smart grid communication. *International Journal of Intelligent Networks* **2024**, *5*, 267–274.
6. Rawte, V.; Sheth, A.; Das, A. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922* **2023**.
7. Cao, M.; Dong, Y.; Cheung, J.C.K. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. *arXiv preprint arXiv:2109.09784* **2021**.
8. Bang, Y.; Cahyawijaya, S.; Lee, N.; Dai, W.; Su, D.; Wilie, B.; Lovenia, H.; Ji, Z.; Yu, T.; Chung, W.; et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023* **2023**.
9. Bang, Y.; Ji, Z.; Schelten, A.; Hartshorn, A.; Fowler, T.; Zhang, C.; Cancedda, N.; Fung, P. Hallulens: Llm hallucination benchmark. *arXiv preprint arXiv:2504.17550* **2025**.
10. Rawte, V.; Chakraborty, S.; Pathak, A.; Sarkar, A.; Tonmoy, S.I.; Chadha, A.; Sheth, A.; Das, A. The troubling emergence of hallucination in large language models-an extensive definition, quantification, and prescriptive remediations. Association for Computational Linguistics, 2023.

 doi:10.20944/preprints202510.0540.v2

11. Orgad, H.; Toker, M.; Gekhman, Z.; Reichart, R.; Szpektor, I.; Kotek, H.; Belinkov, Y. Llms know more than they show: On the intrinsic representation of llm hallucinations. *arXiv preprint arXiv:2410.02707* **2024**.
12. Guan, J.; Dodge, J.; Wadden, D.; Huang, M.; Peng, H. Language models hallucinate, but may excel at fact verification. *arXiv preprint arXiv:2310.14564* **2023**.
13. Huang, Y.; Feng, X.; Feng, X.; Qin, B. The factual inconsistency problem in abstractive text summarization: A survey. *arXiv preprint arXiv:2104.14839* **2021**.
14. Liu, H.; Xue, W.; Chen, Y.; Chen, D.; Zhao, X.; Wang, K.; Hou, L.; Li, R.; Peng, W. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253* **2024**.
15. Chakraborty, N.; Ornik, M.; Driggs-Campbell, K. Hallucination detection in foundation models for decision-making: A flexible definition and review of the state of the art. *ACM Computing Surveys* **2025**, *57*, 1–35.
16. Mündler, N.; He, J.; Jenko, S.; Vechev, M. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *arXiv preprint arXiv:2305.15852* **2023**.
17. 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. *Computational Linguistics* **2025**, pp. 1–45.
18. Le, T.H.; Chen, H.; Babar, M.A. Deep learning for source code modeling and generation: Models, applications, and challenges. *ACM Computing Surveys (CSUR)* **2020**, *53*, 1–38.
19. 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. *arXiv preprint arXiv:2404.00971* **2024**.
20. Bai, Z.; Wang, P.; Xiao, T.; He, T.; Han, Z.; Zhang, Z.; Shou, M.Z. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930* **2024**.
21. Sajjadi, M.; Borhani Peikani, M. The Impact of Artificial Intelligence on Healthcare: A Survey of Applications in Diagnosis, Treatment, and Patient Monitoring **2024**.
22. Mohammadabadi, S.M.S.; Seyedkhamoushi, F.; Mostafavi, M.; Peikani, M.B. Examination of AI's role in Diagnosis, Treatment, and Patient care. In *Transforming gender-based healthcare with AI and machine learning*; CRC Press, 2024; pp. 221–238.
23. Mohammadabadi, S.M.S.; Peikani, M.B. Identification and classification of rheumatoid arthritis using artificial intelligence and machine learning. In *Diagnosing Musculoskeletal Conditions using Artifical Intelligence and Machine Learning to Aid Interpretation of Clinical Imaging*; Elsevier, 2025; pp. 123–145.
24. 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. *arXiv preprint arXiv:2503.05777* **2025**.
25. Banerjee, S.; Agarwal, A.; Singla, S. Llms will always hallucinate, and we need to live with this. *arXiv preprint arXiv:2409.05746* **2024**.
26. Anh, D.H.; Tran, V.; Nguyen, L.M. Analyzing Logical Fallacies in Large Language Models: A Study on Hallucination in Mathematical Reasoning. In Proceedings of the JSAI International Symposium on Artificial Intelligence. Springer, 2025, pp. 179–195.
27. Hao, Y.; Yu, H.; You, J. Beyond Facts: Evaluating Intent Hallucination in Large Language Models. *arXiv preprint arXiv:2506.06539* **2025**.
28. Zhang, Z.; Wang, C.; Wang, Y.; Shi, E.; Ma, Y.; Zhong, W.; Chen, J.; Mao, M.; Zheng, Z. Llm hallucinations in practical code generation: Phenomena, mechanism, and mitigation. *Proceedings of the ACM on Software Engineering* **2025**, *2*, 481–503.
29. Quevedo, E.; Salazar, J.Y.; Koerner, R.; Rivas, P.; Cerny, T. Detecting hallucinations in large language model generation: A token probability approach. In Proceedings of the World Congress in Computer Science, Computer Engineering & Applied Computing. Springer, 2024, pp. 154–173.
30. Kim, S.S.; Liao, Q.V.; Vorvoreanu, M.; Ballard, S.; Vaughan, J.W. " I'm Not Sure, But...": Examining the Impact of Large Language Models' Uncertainty Expression on User Reliance and Trust. In Proceedings of the Proceedings of the 2024 ACM conference on fairness, accountability, and transparency, 2024, pp. 822–835.
31. Xue, Y.; Greenewald, K.; Mroueh, Y.; Mirzasoleiman, B. Verify when uncertain: Beyond self-consistency in black box hallucination detection. *arXiv preprint arXiv:2502.15845* **2025**.
32. Jiang, L.; Jiang, K.; Chu, X.; Gulati, S.; Garg, P. Hallucination detection in LLM-enriched product listings. In Proceedings of the Proceedings of the Seventh Workshop on e-Commerce and NLP@ LREC-COLING 2024, 2024, pp. 29–39.
33. Friel, R.; Sanyal, A. Chainpoll: A high efficacy method for llm hallucination detection. *arXiv preprint arXiv:2310.18344* **2023**.

 doi:10.20944/preprints202510.0540.v2

34. Khadangi, A.; Sartipi, A.; Tchappi, I.; Bahmani, R. Noise Augmented Fine Tuning for Mitigating Hallucinations in Large Language Models. *arXiv preprint arXiv:2504.03302* **2025**.
35. Cheng, M.; Luo, Y.; Ouyang, J.; Liu, Q.; Liu, H.; Li, L.; Yu, S.; Zhang, B.; Cao, J.; Ma, J.; et al. A survey on knowledge-oriented retrieval-augmented generation. *arXiv preprint arXiv:2503.10677* **2025**.
36. Dhuliawala, S.; Komeili, M.; Xu, J.; Raileanu, R.; Li, X.; Celikyilmaz, A.; Weston, J. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495* **2023**.
37. Lavrinovics, E.; Biswas, R.; Bjerva, J.; Hose, K. Knowledge graphs, large language models, and hallucinations: An nlp perspective. *Journal of Web Semantics* **2025**, *85*, 100844.
38. 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. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, Vol. 38, pp. 18126–18134.
39. Dutta, T.; Liu, X. FaCTQA: Detecting and Localizing Factual Errors in Generated Summaries Through Question and Answering from Heterogeneous Models. In Proceedings of the 2024 International Joint Conference on Neural Networks (IJCNN). IEEE, 2024, pp. 1–8.
40. Gu, J.; Jiang, X.; Shi, Z.; Tan, H.; Zhai, X.; Xu, C.; Li, W.; Shen, Y.; Ma, S.; Liu, H.; et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594* **2024**.
41. Li, H.; Dong, Q.; Chen, J.; Su, H.; Zhou, Y.; Ai, Q.; Ye, Z.; Liu, Y. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579* **2024**.
42. Luo, J.; Li, T.; Wu, D.; Jenkin, M.; Liu, S.; Dudek, G. Hallucination detection and hallucination mitigation: An investigation. *arXiv preprint arXiv:2401.08358* **2024**.
43. 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. *arXiv preprint arXiv:2308.11764* **2023**.
44. Manakul, P.; Liusie, A.; Gales, M.J. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896* **2023**.
45. Liu, Y.; Iter, D.; Xu, Y.; Wang, S.; Xu, R.; Zhu, C. G-eval: NLG evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634* **2023**.
46. Song, J.; Wang, X.; Zhu, J.; Wu, Y.; Cheng, X.; Zhong, R.; Niu, C. RAG-HAT: A hallucination-aware tuning pipeline for LLM in retrieval-augmented generation. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, 2024, pp. 1548–1558.
47. Fang, X.; Huang, Z.; Tian, Z.; Fang, M.; Pan, Z.; Fang, Q.; Wen, Z.; Pan, H.; Li, D. Zero-resource hallucination detection for text generation via graph-based contextual knowledge triples modeling. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2025, Vol. 39, pp. 23868–23877.
48. Sun, Z.; Zang, X.; Zheng, K.; Song, Y.; Xu, J.; Zhang, X.; Yu, W.; Li, H. Redeep: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability. *arXiv preprint arXiv:2410.11414* **2024**.
49. Li, N.; Li, Y.; Liu, Y.; Shi, L.; Wang, K.; Wang, H. Drowzee: Metamorphic testing for fact-conflicting hallucination detection in large language models. *Proceedings of the ACM on Programming Languages* **2024**, *8*, 1843–1872.
50. Su, W.; Wang, C.; Ai, Q.; Hu, Y.; Wu, Z.; Zhou, Y.; Liu, Y. Unsupervised real-time hallucination detection based on the internal states of large language models. *arXiv preprint arXiv:2403.06448* **2024**.
51. Wu, S.; Fei, H.; Pan, L.; Wang, W.Y.; Yan, S.; Chua, T.S. Combating Multimodal LLM Hallucination via Bottom-Up Holistic Reasoning. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2025, Vol. 39, pp. 8460–8468.
52. Gunjal, A.; Yin, J.; Bas, E. Detecting and preventing hallucinations in large vision language models. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, Vol. 38, pp. 18135–18143.
53. Sırt, M.; Eyüpoğlu, C. A User-Friendly and Explainable Framework for Redesigning AutoML Processes with Large Language Models. In Proceedings of the 2025 33rd Signal Processing and Communications Applications Conference (SIU). IEEE, 2025, pp. 1–4.
54. Lee, K.; Ippolito, D.; Nystrom, A.; Zhang, C.; Eck, D.; Callison-Burch, C.; Carlini, N. Deduplicating training data makes language models better. In Proceedings of the Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 8424–8445.
55. Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; Van Den Driessche, G.B.; Lespiau, J.B.; Damoc, B.; Clark, A.; et al. Improving language models by retrieving from trillions of tokens. In Proceedings of the International conference on machine learning. PMLR, 2022, pp. 2206–2240.

 doi:10.20944/preprints202510.0540.v2

56. Meng, K.; Bau, D.; Andonian, A.; Belinkov, Y. Locating and editing factual associations in gpt. *Advances in neural information processing systems* **2022**, *35*, 17359–17372.
57. Meng, K.; Sharma, A.S.; Andonian, A.; Belinkov, Y.; Bau, D. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229* **2022**.
58. Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C.D.; Ermon, S.; Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems* **2023**, *36*, 53728–53741.
59. Chuang, Y.S.; Xie, Y.; Luo, H.; Kim, Y.; Glass, J.; He, P. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883* **2023**.
60. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* **2020**, *33*, 9459–9474.
61. Chen, Z.Z.; Ma, J.; Zhang, X.; Hao, N.; Yan, A.; Nourbakhsh, A.; Yang, X.; McAuley, J.; Petzold, L.; Wang, W.Y. A survey on large language models for critical societal domains: Finance, healthcare, and law. *arXiv preprint arXiv:2405.01769* **2024**.
62. Sathyanarayana, S.V.; Shah, R.; Hiremath, S.D.; Panda, R.; Jana, R.; Singh, R.; Irfan, R.; Murali, A.; Ramsundar, B. DeepRetro: Retrosynthetic Pathway Discovery using Iterative LLM Reasoning. *arXiv preprint arXiv:2507.07060* **2025**.
63. Huang, B.; Chen, C.; Xu, X.; Payani, A.; Shu, K. Can Knowledge Editing Really Correct Hallucinations? *arXiv preprint arXiv:2410.16251* **2024**.
64. 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. *Advances in Neural Information Processing Systems* **2023**, *36*, 46534–46594.
65. Shi, W.; Han, X.; Lewis, M.; Tsvetkov, Y.; Zettlemoyer, L.; Yih, W.t. Trusting your evidence: Hallucinate less with context-aware decoding. In Proceedings of the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), 2024, pp. 783–791.
66. Li, K.; Patel, O.; Viégas, F.; Pfister, H.; Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems* **2023**, *36*, 41451–41530.
67. Amatriain, X. Measuring and mitigating hallucinations in large language models: amultifaceted approach, 2024.
68. Rejeleene, R.; Xu, X.; Talburt, J. Towards trustable language models: Investigating information quality of large language models. *arXiv preprint arXiv:2401.13086* **2024**.
69. Li, Y.; Bubeck, S.; Eldan, R.; Del Giorno, A.; Gunasekar, S.; Lee, Y.T. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463* **2023**.
70. Abdin, M.; Aneja, J.; Behl, H.; Bubeck, S.; Eldan, R.; Gunasekar, S.; Harrison, M.; Hewett, R.J.; Javaheripi, M.; Kauffmann, P.; et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905* **2024**.
71. Yang, C.; Zhu, Y.; Lu, W.; Wang, Y.; Chen, Q.; Gao, C.; Yan, B.; Chen, Y. Survey on knowledge distillation for large language models: methods, evaluation, and application. *ACM Transactions on Intelligent Systems and Technology* **2024**.
72. Xiao, T.; Yuan, Y.; Zhu, H.; Li, M.; Honavar, V.G. Cal-dpo: Calibrated direct preference optimization for language model alignment. *Advances in Neural Information Processing Systems* **2024**, *37*, 114289–114320.
73. Xie, Y.; Li, G.; Xu, X.; Kan, M.Y. V-dpo: Mitigating hallucination in large vision language models via vision-guided direct preference optimization. *arXiv preprint arXiv:2411.02712* **2024**.
74. Zhang, Z.; Li, Y.; Kan, Z.; Cheng, K.; Hu, L.; Wang, D. Locate-then-edit for multi-hop factual recall under knowledge editing. *arXiv preprint arXiv:2410.06331* **2024**.
75. Zhang, Z.; Liu, Z.; Patras, I. Grace: A generative approach to better confidence elicitation in large language models. *arXiv preprint arXiv:2509.09438* **2025**.
76. Zheng, C.; Li, L.; Dong, Q.; Fan, Y.; Wu, Z.; Xu, J.; Chang, B. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv:2305.12740* **2023**.
77. Li, N.; Song, Y.; Wang, K.; Li, Y.; Shi, L.; Liu, Y.; Wang, H. Detecting LLM Fact-conflicting Hallucinations Enhanced by Temporal-logic-based Reasoning. *arXiv preprint arXiv:2502.13416* **2025**.
78. Zhang, H.; Deng, H.; Ou, J.; Feng, C. Mitigating spatial hallucination in large language models for path planning via prompt engineering. *Scientific Reports* **2025**, *15*, 8881.

 doi:10.20944/preprints202510.0540.v2

79. Plaat, A.; Wong, A.; Verberne, S.; Broekens, J.; van Stein, N.; Back, T. Reasoning with large language models, a survey. *arXiv preprint arXiv:2407.11511* **2024**.
80. Tang, F.; Huang, Z.; Liu, C.; Sun, Q.; Yang, H.; Lim, S.N. Intervening anchor token: Decoding strategy in alleviating hallucinations for MLLMs. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
81. Lin, S.; Hilton, J.; Evans, O. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958* **2021**.
82. Pan, L.; Saxon, M.; Xu, W.; Nathani, D.; Wang, X.; Wang, W.Y. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188* **2023**.
83. Zheng, L.; Chiang, W.L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems* **2023**, *36*, 46595–46623.
84. Bi, B.; Liu, S.; Wang, Y.; Mei, L.; Gao, H.; Fang, J.; Cheng, X. Struedit: Structured outputs enable the fast and accurate knowledge editing for large language models. *arXiv preprint arXiv:2409.10132* **2024**.