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Not peer-reviewed version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: 8 October 2025

doi: 10.20944/preprints202510.0540.v1

Keywords: hallucination; Large Language Models (LLMs); factuality; faithfulness; hallucination detection; hallucination mitigation; multimodal AI; AI safety

 doi:10.20944/preprints202510.0540.v1

Review

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

Seyed Mahmoud Sajjadi Mohammadabadi [1,2], Burak Cem Kara [1], Can Eyupoglu [1,3] and Oktay Karakuş [4,*]

[1] Battle Born AI, Nevada Center for Applied Research, University of Nevada, Reno, NV 89557, USA
[2] Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA
[3] Department of Computer Engineering, Turkish Air Force Academy, National Defence University, İstanbul 34149, Türkiye
[4] School of Computer Science and Informatics, Cardiff University, Cardiff CF24 4AG, UK
* Correspondence: karakuso@cardiff.ac.uk

Abstract

Hallucination, the generation of plausible yet factually incorrect content, remains a critical barrier to the reliable deployment of Large Language Models (LLMs). This review synthesizes the state-of-the-art in understanding, detecting, and mitigating LLM hallucinations. We begin by establishing a clear taxonomy, tracing the concept's evolution from a broad notion of factual error to a more precise definition centered on unfaithfulness to a model's accessible knowledge. We then survey detection methodologies, categorizing them by model access requirements and examining techniques such as uncertainty estimation, consistency checking, and knowledge grounding. Finally, we provide a structured overview of mitigation strategies organized by their application across the model lifecycle: (1) data-centric approaches like high-quality curation, (2) model-centric alignment through preference optimization and knowledge editing, and (3) inference-time techniques such as Retrieval-Augmented Generation (RAG) and self-correction. We conclude that a layered, "defense-in-depth" strategy is essential for robust mitigation. Key open challenges are scalable data curation, the alignment-capability trade-off, and editing reasoning paths over facts.

Keywords: hallucination; Large Language Models (LLMs); factuality; faithfulness; hallucination detection; hallucination mitigation; multimodal AI; AI safety

1. Introduction

The rapid growth of Large Language Models (LLMs), which have been extensively surveyed for their evolution, architectures, and applications [1] has introduced a paradigm shift in artificial intelligence (AI), marking a key phase in the broader evolutionary roadmap from generative to innovative systems [2] This expansion includes novel applications in specialized fields like education, where AI tools are now used to generate instructional video content for complex topics such as navigating healthcare systems [3]. Yet, their reliability is persistently challenged by the phenomenon of hallucination, the generation of content that appears plausible but is factually incorrect, unfaithful to the provided context, or logically inconsistent [4]. As these models are integrated into increasingly critical domains, from information retrieval and medical diagnostics to complex engineering systems like smart grid communication [5], a precise and robust framework for understanding and categorizing hallucinations has become necessary. This survey begins by reviewing the definitions and taxonomies proposed in the academic literature. It then traces the conceptual evolution from a general notion of factual deviation to a more refined understanding that distinguishes a model's internal consistency from its alignment with external reality, offering a unified perspective crucial for developing effective detection methodologies [6].

 doi:10.20944/preprints202510.0540.v1

Table 1. Summary of Hallucination and Factuality Concepts in LLMs.

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

1.1. Definition: From Factual Deviation to Internal Inconsistency

The definition of hallucination describes it as a model-generated output that drifts from factual reality or includes fabricated information. This initial conceptualization was sufficient for early-generation models, where the primary concern was the generation of blatant falsehoods. However, as LLMs have grown in complexity and capability, this definition has proven to be too broad, conflating two distinct types of model failure: a failure to be correct about the world (factuality) and a failure to be consistent with its own accessible knowledge (hallucination) [7]. Recent work has sought to disentangle these concepts by proposing a clearer taxonomy. Notably, Bang et al. define hallucination as output that contradicts the model's accessible knowledge—its training data and inference context—while factuality refers to consistency with real-world, verifiable facts external to the model [8]. This distinction is more than semantic; it underpins how hallucinations are detected and mitigated. A model may produce factually incorrect output that aligns with its training data (not a hallucination) or factually correct output that contradicts a provided source (an intrinsic hallucination). Hallucination thus concerns consistency with known or provided context, while factuality refers to correctness against external reality. Recognizing this separation enables more targeted detection: internal consistency checks differ fundamentally from external fact-checking [9].

1.2. Intrinsic vs. Extrinsic

Building on this foundational distinction, the literature has developed two primary types to classify hallucinations: the intrinsic/extrinsic framework and the factuality/faithfulness framework [9].

- **Intrinsic Hallucinations**: An intrinsic hallucination occurs when the model generates content that directly contradicts the user-provided input. It reflects a failure to remain faithful to the immediate context. For example, if a source text states that a company's revenue was $10 million, but the summary produced by the model reports $15 million, this constitutes an intrinsic hallucination [10]. This category focuses on the model's ability to accurately represent information available during inference [11].
- **Extrinsic Hallucinations**: This error occurs when the model generates content that cannot be verified from the provided context and is inconsistent with its training data. It involves fabricating

 doi:10.20944/preprints202510.0540.v1

unsupported information. For example, if a summary includes, "The CEO also announced plans to expand into the Asian market," despite no such statement in the source or likely in training data, it constitutes an extrinsic hallucination. This category reflects the model's tendency to over-extrapolate or fill gaps beyond its knowledge, often when generating novel content based solely on task instructions. It underscores the model's limitations in recognizing the boundaries of its internal knowledge [10].

1.3. Factuality vs. Faithfulness Framework

An alternative taxonomy introduced by Huang et al. [4] classifies hallucinations based on the principle violated: adherence to user-provided context (faithfulness) versus alignment with objective external facts (factuality). While insightful, this framework can blur the line between internal consistency and factual correctness.

It is important to distinguish hallucination from low factuality, though they often overlap. **Hallucination** concerns whether the model's output is consistent with its accessible knowledge—i.e., the input context or training data—highlighting issues of internal coherence. In contrast, **factuality** assesses the objective correctness of the content relative to external, real-world information [7,12]. A model may produce an internally consistent response that is nonetheless factually inaccurate due to outdated training data or changes in the real world.

Hallucinated content often appears fluent, confident, and persuasive, complicating detection for both automated tools and human users. While some argue that controlled hallucination may be desirable in creative applications, this is not the case in high-stakes domains such as medicine, law, or finance. In these settings, mitigating hallucinations is essential to ensure 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 Multimodal Large Language Models (MLLMs) denote a discrepancy between factual visual content and the corresponding generated textual output. These hallucinations may appear either as judgment deficiencies (e.g., incorrect true/false responses) or descriptive inaccuracies (e.g., mismatched visual details) [14]. As LLMs are increasingly deployed in diverse modalities and applications, a finer-grained taxonomy has emerged to address the distinct forms of hallucinations that occur across different domains.

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

  Factual contradiction occurs when the model generates statements that are inconsistent with real-world knowledge [4]. These may include entity-error hallucinations (e.g., naming an incorrect entity) or relation-error hallucinations (e.g., misrepresenting a relationship between entities).
  Context-conflicting hallucinations emerge when generated output contradicts previous model outputs. For example, a summary might incorrectly substitute a person's name mentioned earlier [17].

 doi:10.20944/preprints202510.0540.v1

Input-conflicting hallucinations arise when generated content diverges from the user's original input, such as adding details not found in the source material [17].
In code generation, *context inconsistency* refers to contradictions between newly generated code and prior code or user input, often manifesting as subtle logical flaws [18,19].
A distinct subcategory is *self-contradiction*, where the model outputs two logically inconsistent statements within the same response, even though it was conditioned on a unified context [19].

- **Fabrication:** This form of hallucination involves the generation of entities, events, or citations that are entirely fictitious or unverifiable [20].

  In MLLMs, fabrication often manifests as mismatches between visual inputs and textual descriptions.
  Object category hallucinations involve naming objects that do not appear in the image (e.g., mentioning a "laptop" or "small dog" that isn't present).
  Object attribute hallucinations occur when properties such as shape, color, material, or quantity are inaccurately described—e.g., referring to a man as "long-haired" when he is not. These may include event misrepresentations or counting errors.
  Object relation hallucinations involve incorrect assertions about spatial or functional relationships (e.g., "the dog is under the table" when it is beside it) [14].
  In high-stakes medical applications, where AI is increasingly 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 broadly, fabrication occurs when LLMs are prompted with questions beyond their training knowledge or lack sufficient data, leading them to generate plausible-sounding but unsupported content.

- **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*, where generated segments serve no purpose and may cause execution failures [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 crucial step toward ensuring the factual reliability of large language models. As summarized in **Table 2**, approaches can be broadly categorized based on the level of access required to the underlying model: white-box, grey-box, and black-box [15]. Each category supports a range of techniques with different trade-offs in accuracy, interpretability, and deployment feasibility.

 doi:10.20944/preprints202510.0540.v1

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 hallucinations	INSIDE (EigenScore), SAPLMA, OPERA, DoLa	Direct mechanistic signal	Requires full model access; model-specific
Grey-box	Uncertainty Quantification	Low confidence = likely hallucination	Token/sequence probability, Semantic entropy, LLM-Check	Efficient; accessible via logits	Probabilistic assumptions may fail; needs API logits access
Black-box	Consistency Checking	Hallucinations vary across outputs	SelfCheckGPT, LM-vs-LM, Multi-Agent Debate	Model-agnostic; any API-accessible model	Expensive (multiple calls); fails if model consistently hallucinates

2.1. Uncertainty-Based Detection

These methods estimate whether hallucinations are likely based on the uncertainty or inconsistentness of the model when generating output.

- **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. A statement that is not consistently supported across multiple generations is flagged as a potential hallucination. However, this method's primary drawback is its computational cost, as it requires multiple inference calls for a single detection, and it may fail if the model consistently makes the same factual error across 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. A typical RAG-based verifier works by first decomposing the generated text into a set of verifiable claims or facts. Each claim is then used as a query to a retriever, which fetches relevant snippets from a knowledge corpus (e.g., Wikipedia or a domain-specific database). Finally, a verifier model, often an NLI (Natural Language Inference) model or another LLM, assesses whether the retrieved evidence supports, refutes, or is neutral towards the claim. While powerful, this method is

 doi:10.20944/preprints202510.0540.v1

dependent on the quality of both the retriever and the external knowledge base; retrieval failures or outdated sources can lead to incorrect verification. [35].

- **Internal Verification (Chain-of-Verification):** This black-box technique cleverly prompts the model to critique its own outputs without external tools. The process unfolds in a structured sequence: (1) the LLM generates an initial response; (2) it is then prompted to devise a series of verification questions to fact-check its own response; (3) it attempts to answer these questions independently; and (4) finally, it generates a revised, final answer based on the outcome of its self-verification process. This method encourages a form of self-reflection, forcing the model to re-examine its claims and correct inconsistencies. Its main advantage is its resource independence, but its effectiveness is bounded by the model's own internal knowledge and its ability to formulate useful verification questions. [36].
- **Knowledge Graph Validation:** Uses structured KGs to cross-check relationships or facts asserted by the model, particularly effective for relational consistency [37,38].
- **ChainPoll Adherence (Closed-domain):** Judges how well output aligns with provided evidence passages using multi-round CoT prompting and majority voting [33].

2.3. Dedicated Detection Models

These methods involve training additional models specialized for hallucination detection.

- **QA-Based Fact Checking:** Converts model outputs to questions and uses QA pipelines to retrieve answers from source text. Discrepancies reveal hallucinations [39].
- **LLM-as-a-Judge:** Prompts a strong LLM (e.g., GPT-4) to evaluate the correctness of another LLM's output based on CoT reasoning. ChainPoll Correctness and G-Eval are examples [40,41].
- **Supervised Classifiers:** Train models directly on hallucinated vs. non-hallucinated examples [42] (e.g., HALOCHECK [43]).

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.v1

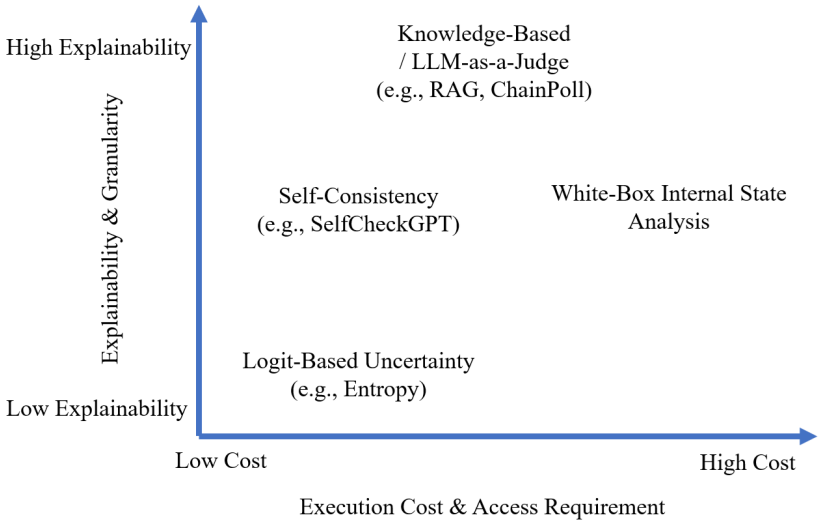

Figure 1. A conceptual landscape of hallucination detection methods, mapping techniques based on their trade-offs between execution cost/access requirements and the explainability of their outputs. Methods in the top-left are highly explainable and accessible via standard APIs, while methods on the right require privileged model access or high computational cost.

2.4. Summary and Practical Considerations

The selection of an appropriate hallucination detection method involves navigating the trade-offs between cost, access, and explainability, as visualized in Figure 1. Key factors guiding this choice include:

- **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:** Whether human-readable justifications are provided (e.g., CoT in ChainPoll, entailment evidence in NLI). This aligns with a growing demand for user-friendly and explainable frameworks in AI-driven processes, such as those being developed for AutoML [53].

As LLMs become more widely deployed, hallucination detection methods must be scalable, reliable across diverse tasks, and adaptable to evolving model architectures.

3. Hallucination Mitigation Strategies

Following the identification and detection of hallucinations, the next critical challenge is their mitigation. Generating plausible yet factually incorrect or ungrounded content remains one of the most significant obstacles to the safe and widespread deployment of LLMs. Mitigation is crucial in high-stakes domains, including medicine, finance, and law [54]. Mitigation is not a single intervention but a multifaceted process that must be addressed across the model lifecycle. Effective strategies are essential to ensure reliability, safety, and ethical use.

The following hallucination mitigation techniques 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 grounding of the data used to train LLMs; (2) **Model-Centric Strategies**, which modify model parameters via fine-tuning and alignment to promote truthful behavior; and (3) **Inference-Time Strategies**, which introduce post hoc mechanisms during generation to reduce hallucinations [4]. These categories are not mutually exclusive; rather, robust solutions often integrate methods from multiple stages to form a layered defense.

Table 4 summarizes this taxonomy, highlighting key principles, representative techniques, and associated trade-offs.

 doi:10.20944/preprints202510.0540.v1

Table 4. 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 [55]).	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 [56,57], MEMIT [58]).	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 [59]); decoding control (DoLa [60], CAD [61]); ITI [62].	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 [63]. 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 [64].

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 enriched with commonsense reasoning and factual content [65,66]. Data augmentation techniques—particularly those ensuring topic diversity and structural coherence—are also key during fine-tuning.

In high-stakes domains such as healthcare and law, expert-curated, domain-specific corpora (e.g., MedCPT [24], verified legal documents) are essential. Rigorous data governance practices, including dataset versioning and prompt logging, further enhance reliability.

3.1.2. Retrieval-Augmented Pre-Training

A more advanced approach incorporates retrieval into the pre-training process itself, enabling models to condition generation on external sources of truth from the outset.

A canonical example is the Retrieval-Enhanced Transformer (RETRO), which augments an LLM with a retrieval module to access a large corpus of supporting documents during training. RETRO consistently outperforms similarly sized non-retrieval models in factuality.

 doi:10.20944/preprints202510.0540.v1

However, this strategy is resource-intensive—raising training costs by up to 25%, and inherits limitations from the retrieval corpus. If the external memory contains outdated or inaccurate information, these errors may become embedded in the model's parameters.

Ultimately, the principle of 'garbage in, garbage out' applies acutely on the LLM scale. Models trained on flawed corpora require inference-time strategies to counteract baked-in inaccuracies, adding latency and complexity. High-quality data remains the most impactful investment in hallucination mitigation.

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

Model-centric strategies modify parameters post-pre-training to instill factually grounded behaviors. They occupy a middle ground—more flexible than rigid data curation, yet more durable than inference-only fixes. We focus on three paradigms: 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.

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.

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 [67].
- **V-DPO**: incorporates visual context to reduce hallucinations in vision-language models [68].

These methods collectively highlight the growing sophistication of model alignment techniques and their importance in mitigating hallucinations in deployed systems.

The rapid evolution from RLHF to DPO reflects a significant macro-trend in LLM alignment: the shift from complex, multi-stage, and often unstable pipelines toward simpler, more efficient alternatives. RLHF typically involves several sequential steps—supervised fine-tuning, preference data collection, reward model training, and reinforcement learning—each of which introduces implementation challenges and hyperparameter sensitivity.

DPO's key innovation is to eliminate the intermediate reward modeling step by reframing alignment as a direct optimization over preference pairs, akin to a classification objective. This simplification retains or even improves performance, suggesting that the benefits of complex, bio-

 doi:10.20944/preprints202510.0540.v1

inspired cognitive processes can often be captured through more mathematically direct formulations. The broader implication is a growing emphasis on scalability and engineering pragmatism in alignment research.

This trajectory points toward a future where alignment objectives may be incorporated earlier in the training pipeline—potentially through *preference-aware pre-training*—reducing reliance on costly post-hoc interventions and integrating alignment more naturally into model development.

3.2.3. Knowledge Editing: Surgical Model Updates

Knowledge editing offers a highly targeted alternative to full fine-tuning, enabling precise modifications to an LLM's internal knowledge without retraining or risking catastrophic forgetting. These methods aim to update or correct individual facts (e.g., changing a CEO's name or a capital city) while preserving the rest of the model's behavior.

Most techniques follow a *locate-then-edit* paradigm [69]. Using mechanistic interpretability tools—such as causal tracing—these methods identify specific MLP layers or neurons responsible for storing a factual association. Once located, they directly modify the corresponding weights to reflect updated information.

Prominent examples include:

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

While effective for single-hop factual updates, locate-then-edit methods struggle with **multi-hop reasoning**. Research shows that shallow layers often store atomic facts, while deeper layers handle reasoning chains and indirect inferences. Consequently, editing a shallow-layer fact (e.g., "Paris is the capital of France") may not update deeper reasoning pathways needed to answer questions like "What is the capital of the country where the Eiffel Tower is located?"

Newer methods such as **IFMET** [69] (Interpretability-based Factual Multi-hop Editing in Transformers) aim to address this by identifying and updating both shallow and deep layers, improving generalization to multi-step reasoning tasks. However, no single editing method currently excels in all key criteria: effectiveness, generalization, location, and performance, and the model architecture and domain remain highly dependent on the model.

The locate-then-edit approach reflects a powerful but limited metaphor: treating knowledge in LLMs as editable "code." Techniques like ROME and MEMIT lend empirical support to this view, showing that factual associations are often localized and editable. However, their failure in multihop reasoning [70] reveals the boundaries of the metaphor. Updating one "line of code" (e.g., "A is the parent of B") does not recompile the full "program" to reflect downstream deductions (e.g., "A is the grandparent of C").

This suggests that factual storage and factual reasoning are distinct and entangled processes within transformer models. Moving from "fact editing" to *reasoning path editing* [71] is a frontier for future work—one that will require more advanced interpretability tools capable of tracing and altering full 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

 doi:10.20944/preprints202510.0540.v1

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 [72].
3. **Advanced Decoding and Intervention:** Modify the decoding process or directly intervene in the model's internal activations [73].

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:** The model generates a draft, then retrieves evidence to revise its own output.

While powerful, Retrieval-Augmented Generation (RAG) introduces new failure points, collectively termed RAG-induced hallucinations, which can manifest at various stages of the pipeline. The process often 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.

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.

 doi:10.20944/preprints202510.0540.v1

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

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.

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 [60].
- **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 [61].

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 seeks to steer the model's internal state toward more truthful representations. ITI begins by applying linear probing to a dataset such as TruthfulQA [74], identifying a sparse set of attention heads whose activations correlate strongly with truthful responses. Then, during inference, a small learned vector is added to the output of these attention heads at each generation step. This nudges the model's activations in truth-correlated directions, substantially improving truthfulness with minimal computational overhead and no 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").

 doi:10.20944/preprints202510.0540.v1

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.
- **Editing Reasoning, Not Just Facts:** Most knowledge editing methods target discrete facts. However, true robustness requires the ability to trace and modify the deeper reasoning paths through which those facts influence behavior—a challenge that demands more advanced 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.

4. Recent Trends and Open Issues

The field of LLM hallucination research is rapidly evolving, moving from foundational definitions toward more sophisticated, integrated, and scalable solutions. While significant progress has been made in identifying and mitigating hallucinations, several key trends and persistent open problems now define the research frontier. This section outlines the most prominent emerging trends in detection and mitigation, alongside the unresolved challenges that will shape future work.

4.1. Emerging Trends in Mitigation and Detection

Current research reflects a clear shift towards more efficient, targeted, and system-level approaches to managing hallucination. Four major trends are apparent:

- **Shift Toward Simpler Alignment Methods:** There is a notable trend away from complex, multi-stage alignment pipelines like Reinforcement Learning from Human Feedback (RLHF) towards simpler, more stable, and mathematically direct 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 relying on costly full-model fine-tuning, researchers are increasingly developing "surgical" methods to edit a model's internal knowledge directly. Techniques like **ROME** and **MEMIT** use mechanistic interpretability to locate and precisely modify the model parameters responsible for storing specific facts. This allows for the efficient correction of factual errors without risking the catastrophic forgetting of other knowledge.
- **Advanced Inference-Time Interventions:** A major area of innovation is the development of techniques that operate during the generation process, requiring no parameter updates. These methods are highly practical as they can be applied to any model, including proprietary APIs. This trend includes **Retrieval-Augmented Generation (RAG)** pipelines that ground outputs in external evidence, **self-correction routines** [75] like Chain-of-Verification (CoVe) that prompt a model to fact-check its own statements, and **advanced decoding strategies** (e.g., DoLa, ITI) that intervene in a model's internal activations to steer it toward more truthful outputs.
- **Development of LLM-as-a-Judge Paradigms:** For detection, there is a growing reliance on using powerful LLMs themselves as evaluators. The **LLM-as-a-Judge** [40,76] paradigm, seen in methods like G-Eval [45], leverages the advanced reasoning capabilities of frontier models (e.g.,

 doi:10.20944/preprints202510.0540.v1

GPT-4) to assess the factuality and faithfulness of outputs from other models, often outperforming traditional metrics and approaching human-level judgment.

4.2. Key Open Problems

Despite significant progress, several foundational challenges remain at the forefront of hallucination research:

- **Scalable and High-Quality Data Curation:** The principle of 'garbage in, garbage out' remains a fundamental obstacle. Developing fully automated, scalable, and multilingual pipelines for curating high-quality, factually accurate, and diverse training data is a critical and unresolved challenge, especially at the scale of trillions of tokens.
- **The Alignment-Capability Trade-off:** A critical open problem is the risk of an "alignment tax," where interventions to enhance factuality and reduce hallucinations inadvertently diminish other valuable model capabilities. Aggressive fine-tuning or preference optimization (like RLHF or DPO) can make a model overly cautious, causing it to refuse to answer reasonable questions or lose its ability to perform complex, multi-step reasoning. For instance, a model heavily optimized for factuality might become less creative or struggle with tasks that require nuanced inference beyond explicitly stated facts. This trade-off forces a difficult balance: how can we make models more truthful without making them less useful? Future research must focus on developing alignment techniques that are more targeted and less disruptive. This could involve disentangling different model capabilities at a mechanistic level, allowing for surgical interventions that correct for factuality without impairing reasoning or creativity. Another promising direction is developing alignment methods that explicitly reward nuanced behaviors, such as expressing uncertainty or providing conditional answers, rather than simply penalizing any output that cannot be externally verified. Synthesizing factuality with utility remains a central challenge for the next generation of LLMs.
- **Editing Reasoning Paths, Not Just Facts:** Current knowledge editing techniques (e.g., STRUEDIT [77]) are effective at correcting discrete, single-hop facts (e.g., "Paris is the capital of France"). However, they struggle to update the complex, multi-hop reasoning pathways that depend on those facts. Advancing from fact-editing to reasoning-path editing is a major frontier that will require more sophisticated mechanistic interpretability tools.
- **Compositionality of Mitigation Techniques:** The most robust systems will likely employ a "defense-in-depth" strategy that layers multiple mitigation techniques. However, the interactions between these methods are poorly understood. Research is needed to develop a principled understanding of how different strategies (e.g., RAG, DPO, and knowledge editing) can be composed effectively without interfering with one another.

5. Conclusions

This review has provided a structured overview of the multifaceted challenge of hallucination in Large Language Models, charting the course from fundamental definitions to advanced detection and mitigation strategies. Our analysis highlights the critical conceptual shift from viewing hallucinations as simple factual errors to understanding them as failures of faithfulness to a model's known context, distinct from external factuality. We categorized the diverse landscape of detection methods by their model access requirements, revealing a fundamental trade-off between the high accuracy of white-box, interpretability-based approaches and the broad applicability of black-box, consistency-based techniques.

In surveying mitigation strategies, we found no single solution; instead, the most effective path forward lies in a layered, "defense-in-depth" approach. This involves integrating techniques across the entire model lifecycle: starting with high-quality, factually grounded data curation (data-centric), followed by robust model alignment using preference optimization and surgical knowledge editing (model-centric), and concluding with real-time, evidence-based grounding at deployment (inference-

 doi:10.20944/preprints202510.0540.v1

time). Despite significant progress, critical challenges remain. Future work must address scalable data curation, the so-called "alignment tax" where mitigation can suppress model capabilities, and the complex frontier of editing not just facts, but the underlying reasoning paths that produce them. Ultimately, managing hallucination is not about its complete elimination but about building a robust, multi-layered ecosystem of tools and practices that ensure LLMs are reliable, transparent, and safe for widespread societal deployment.

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.

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.

 doi:10.20944/preprints202510.0540.v1

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.
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**.

 doi:10.20944/preprints202510.0540.v1

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**.
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.

 doi:10.20944/preprints202510.0540.v1

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. 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**.
55. 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**.
56. Huang, B.; Chen, C.; Xu, X.; Payani, A.; Shu, K. Can Knowledge Editing Really Correct Hallucinations? *arXiv preprint arXiv:2410.16251* **2024**.
57. 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.
58. Meng, K.; Sharma, A.S.; Andonian, A.; Belinkov, Y.; Bau, D. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229* **2022**.
59. 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.
60. 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**.
61. 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.
62. 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.
63. Amatriain, X. Measuring and mitigating hallucinations in large language models: amultifaceted approach, 2024.
64. Rejeleene, R.; Xu, X.; Talburt, J. Towards trustable language models: Investigating information quality of large language models. *arXiv preprint arXiv:2401.13086* **2024**.
65. 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**.
66. 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**.
67. 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.
68. 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**.

 doi:10.20944/preprints202510.0540.v1

69. 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**.
70. 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**.
71. 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.
72. 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**.
73. 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.
74. Lin, S.; Hilton, J.; Evans, O. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958* **2021**.
75. 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**.
76. 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.
77. 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**.