Managing hallucination risk in LLM deployments at the EY organization

The better the question. The better the answer.
The better the world works.

Shape the future with confidence

Table of Contents

About the authors 02

Executive summary 03

1. Introduction 04

2. Problem statement 05

3. Background and related work 09

4. Hallucination mitigation methods and strategies 10

5. Proposed target state operating model 20

References 21

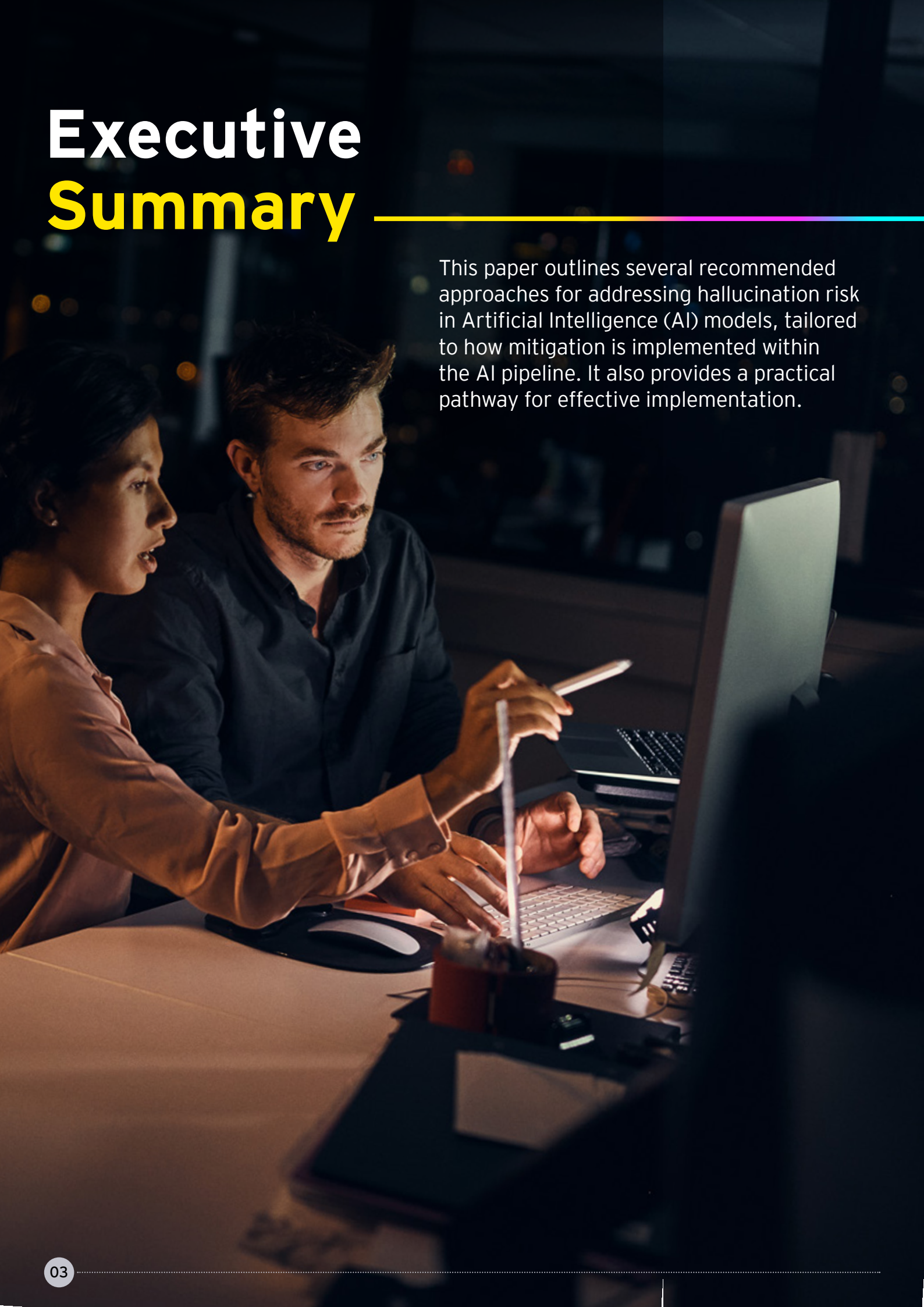
Appendix 1. Essential governance tools 24

About the Authors

Niloufar Shoeibi, Jonathan DeGange and Kavitha Elangovan lead the EY Client Technology Artificial Intelligence (AI) Quality Risk Management function, where they are responsible for risk assessments and third-party AI model reviews. Mauricio Pommier Gasser is the leader of EY Client Technology AI competency, overseeing AI deployments across service lines and member firms. All authors work at the global EY organization (EY) in the Client Technology Engineering (CTE) AI competency.

Executive Summary

This paper outlines several recommended approaches for addressing hallucination risk in Artificial Intelligence (AI) models, tailored to how mitigation is implemented within the AI pipeline. It also provides a practical pathway for effective implementation.



Chapter 01

Introduction

Large language models (LLMs) are rapidly transforming service delivery and internal operations at the global EY organization (EY), creating new opportunities for efficiency, insight and innovation. At the same time, they introduce the critical challenge of hallucinations – instances where models generate factually incorrect or misleading information. In high-risk domains such as tax, audit and other services, these risks can have serious consequences that can impact compliance, client trust and reputation of the EY organization.

Hallucinations are not unique to LLMs or LLM Agents [1], however, throughout this paper, we are mainly interested in LLMs, AI Agents or combinations thereof. In the context of LLMs, hallucinations refer to outputs that are factually incorrect, fabricated or misleading yet presented with high confidence [2]. In the context of EY, hallucinations in large language models (LLMs) can manifest in critical deliverables such as audit reports, tax compliance guidance, due diligence assessments or risk advisory outputs – where even minor factual inaccuracies may lead to significant financial, reputational or regulatory consequences. This risk is especially pronounced in professional services domains like tax, legal and consulting, where an LLM that fabricates a regulation, misstates an accounting principle or invents a legal precedent could compromise compliance, erode client trust and expose the global EY organization to serious reputational and regulatory liabilities.

The significance of hallucination risk is amplified in 2025, as organizations increasingly adopt generative AI for high-value, high-stakes use cases [3]. Regulatory frameworks such as the EU AI Act [4], alongside industry standards for governance and compliance, underscore the need for enterprises like EY to demonstrate proactive management of AI risks.

Recognizing these challenges, hallucination risk is considered alongside AI reliability, safety and ethical dimensions, confirming mitigation is holistic, not purely technical. This paper outlines the nature of hallucinations in LLMs, their implications for the global EY organization and other firms, and practical mitigation strategies for deployment at scale. The aim is to provide a clear, actionable roadmap that aligns with EY governance framework, protects compliance and reinforces client trust.

Chapter 02

Problem statement

In large language models (LLMs), hallucination refers to outputs that appear coherent and authoritative but lack grounding in verified knowledge or accurate data. Although syntactically fluent, such responses may be factually incorrect, fabricated or misleading. In compliance-sensitive, client-facing and decision-critical contexts, hallucinations introduce material risks that can undermine trust and operational integrity [5].

The business impact is significant: hallucinated outputs can mislead audit teams, compromise advisory deliverables, expose organizations to regulatory scrutiny, damage reputations and erode internal confidence in AI-enabled tools [6]. As enterprise deployments scale in 2025, both clients and regulators are demanding demonstrable safeguards. Addressing hallucination risk is no longer optional – it is a strategic imperative for delivering reliable, compliant and trusted AI services.

Internal vs external hallucinations

Hallucinations fall into two broad categories. **Intrinsic hallucinations** arise from internal reasoning errors or limitations in the training distribution, even when the inputs are accurate. **Extrinsic hallucinations**, by contrast, misstate or fabricate external facts, entities or sources due to missing, outdated or insufficient grounding (5). Table 1 provides more details on internal versus external hallucinations.

Table 1

Internal vs external hallucinations

Feature	Internal hallucination	External hallucination
Example	When asked "Who was the 44th President of the United States?", a model trained before 2008 might confidently answer "Richard Nixon," ignoring more current information present in its broader dataset.	A user provides an article stating the FDA approved the first Ebola vaccine in 2019. An LLM, when asked to summarize, generates a text stating the FDA rejected it. This contradicts the provided source context.
Origin of error	The model misinterprets or fabricates information from its own vast, pre-trained knowledge base, which can contain flaws, biases or outdated facts. The inconsistency is purely internal to the model's learned data.	The model generates information that is inconsistent with or cannot be verified against specific, verifiable source context provided by the user.
Conflicts with	The model's own "world knowledge," which is encoded in its parameters during training.	The specific input documents or "source material," provided to the model during a particular query, such as in a retrieval-augmented generation (RAG) system.
Common causes	**Outdated knowledge:** Information learned during pre-training is no longer current. **Overfitting:** The model memorizes patterns too rigidly and fails to generalize correctly. **Limited reasoning:** The model fails to follow a correct logical chain, especially in complex tasks.	**Instruction inconsistency:** The model ignores or misunderstands explicit user instructions. **Context inconsistency:** The model adds facts not present in the source text or contradicts information given in the prompt. **Insufficient retrieval:** Retrieval step fails to provide the LLM with the correct document excerpts.

Types of hallucinations

Within intrinsic and extrinsic hallucination categories, we derive eight subcategories of hallucinations. We discuss these here:

1. **Inconsistent answers.** Answers provided by the LLM are inconsistent upon repeat inference. This issue is linked to the intrinsic model architecture.
2. **Overconfident tone.** The LLM claims it is "sure" an answer is correct, deceiving the user into accepting the output, but the output is factually incorrect.
3. **Wrong numbers or values in extraction tasks.** The most often-cited type of hallucination, this is when the output of the LLM is "incorrect" or "gave the wrong answer".
4. **Unsupported outputs in knowledge-related tasks.** Here, the LLM claims percentages or totals with no source data and appears to be "made up".
5. **Misinterpreted policy.** Here the LLM executes the system prompt incorrectly, including ignoring exceptions or specific directions provided by the user.
6. **Fabricated entries.** Here the AI system provides nonexistent entities, transactions or facts.
7. **Outdated references.** Here the LLM is using stale knowledge data, yet the LLM itself has no issues.
8. **Invented references (citations) in knowledge tasks.** Here the LLM generates synthetic or fake citation references.

Below we provide a detailed table of common manifestations of hallucination in enterprise contexts, illustrating how these risks materialize across operational workflows and where they are discussed in each section of the paper.

Table 3

Enterprise AI hallucination patterns and mitigations.

Manifestation	Brief definition	Taxonomy tag (Huang et al.) or Type	Risk or impact	Primary mitigations
Invented citations or sources	References that don't exist or don't support the claim	Factuality or Extrinsic	Reputational and legal exposure	RAG over authoritative corpora. Provenance enforcement. Block unsourced claims. **Hunan in the Loop (HITL)**

Manifestation	Brief definition	Taxonomy tag (Huang et al.) or Type	Risk or impact	Primary mitigations
Out-of-date references (contextual drift)	Uses superseded versions of rules or standards	Factuality or Extrinsic	Non-compliance, rework	§4.1 Versioned corpora and validity windows, freshness checks, §4.8 change control
Misinterpretation of policy or regulation	Reads context incorrectly or ignores exceptions	Faithfulness – context or Intrinsic+ Extrinsic	Flawed guidance, client risk	§4.1 Evidence-aware RAG, §4.5 claim verifiers, §4.2.2 4.8 HITL, §Knowledge graph
Fabricated entities or events	Nonexistent company, transaction or precedent	Factuality or Extrinsic	Selection errors; auditability gaps	§4.5 Entity or KG checks, §4.1 KG-augmented retrieval, abstention
Numeric fabrication or miscalculation	Invented numbers or wrong math	Faithfulness – Logical or Intrinsic	Bad decisions; financial errors	§4.4 Tool-use (calc/SQL), guards & unit tests, §4.5 auto checks
Overconfident tone without qualifiers	Presents uncertain outputs as certain	Faithfulness – Instruction or Intrinsic	Over-trust, decision risk	§4.5 Calibration & confidence. §4.8 abstention policy
Inconsistent answers to similar prompts	Contradictory outputs across near-duplicates	Faithfulness – Logical or Intrinsic	Trust erosion, rework	§4.2 Self-consistency + critique; canonical prompt templates
Unsupported generalizations (“hallucinated stats”)	Claims percentages or totals with no source	Factuality or Extrinsic	Misleading narratives	§4.1 Provenance requirement, retrieve-then-generate, §4.8 block unsourced
Misattribution or wrong authorship or source	Correct fact, wrong source or author	Factuality or Extrinsic	Credibility loss	§4.5 Entity linking and source validation, §4.1 Rerank by source authority
Nonexistent attachments orIDs	Refers to files orIDs that aren’t present	Faithfulness – Context orExtrinsic	Client friction, incidents	§4.8 Pre-send validators, ID-bound templates, abstain if missing

Chapter 03

Background and related work

3.1 Survey literature and taxonomies

Recent surveys have provided systematic overviews of hallucinations in large language models. In this research [6], a comprehensive taxonomy distinguishes hallucinations by principles of factuality and faithfulness, framing them as either intrinsic (arising from model reasoning or parametric memory) or extrinsic (errors due to misrepresentation of external facts). Subsequent studies built on this taxonomy to explore domain-specific contexts, reflecting the diverse risks across industries.

3.2 Domain-specific risks

In healthcare, hallucinations pose particularly high stakes. In this research [7], explainability methods are integrated into clinical language models to detect fabricated or unsupported recommendations. Other studies [8] document healthcare professionals' concerns about generative AI scribes introducing inaccurate or biased clinical notes. Provenance-aware guardrails are also emphasized as necessary in medical applications [9].

In finance, hallucinations appear in fabricated tabular values or unsupported metrics. A framework to systematically assess hallucinations in financial LLM outputs is proposed in [10], while another generative framework for verifiable financial record summarization is introduced in [11]. These studies highlight the importance of explainability and validation for compliance-heavy domains.

In education and learning support, the effects of hallucinations on scaffolding tasks for students are examined in [12], with mitigation strategies such as retrieval-augmented generation (RAG) suggested to enhance factual reliability without stifling adaptive learning benefits.

3.3 Mitigation approaches in literature

Across domains, three families of mitigation approaches are emerging. Retrieval augmented generation (RAG) is a widely studied and deployed method for reducing hallucinations [5]. Explainability and provenance enforcement, particularly in clinical and financial contexts, are increasingly recognized as prerequisites for deployment [7, 11]. Fine tuning strategies, such as preference optimization and hallucination aware supervised training, are applied to improve model faithfulness [10].

3.4 Debates on hallucination utility

While many research treats hallucination as a risk, some studies note its potential creative value. Generative models' tendency to fabricate can be reframed as a feature in exploratory or creative contexts [13]. Hallucinations are also discussed in relation to cognitive semantics, suggesting productive uses in brainstorming and cultural interpretation [14].

3.5 Gap analysis

Despite this growing literature, gaps remain. Domain-specific mitigation methods are not yet well aligned with enterprise governance frameworks, such as the EU AI Act and few studies integrate hallucination risk with broader compliance systems. Scalability remains a challenge, as human-in-the-loop verification does not scale to enterprise deployment [15]. Dynamic knowledge alignment, particularly in regulatory and financial domains, is still underexplored. The EY approach contributes by explicitly integrating hallucination risk management into institutional AI governance, bridging the technical, compliance and cultural dimensions of mitigation.

Chapter 04

Hallucination mitigation methods and strategies

This section organizes mitigation into two complementary families inspired by recent surveys [6] [12] and taxonomies: **prompt-side controls** (methods that act at inference or prompt time) and **model-development controls** (training or decoding-time methods). We end with cross-cutting detection, provenance and governance. Throughout, "hallucination" is framed using the factuality or faithfulness taxonomy (factual conflicts with reality and faithfulness deviations from instruction, context or logic). Figure 1 summarizes the mitigation space we follow in this section, grouping techniques into prompt-side controls and model-development controls.

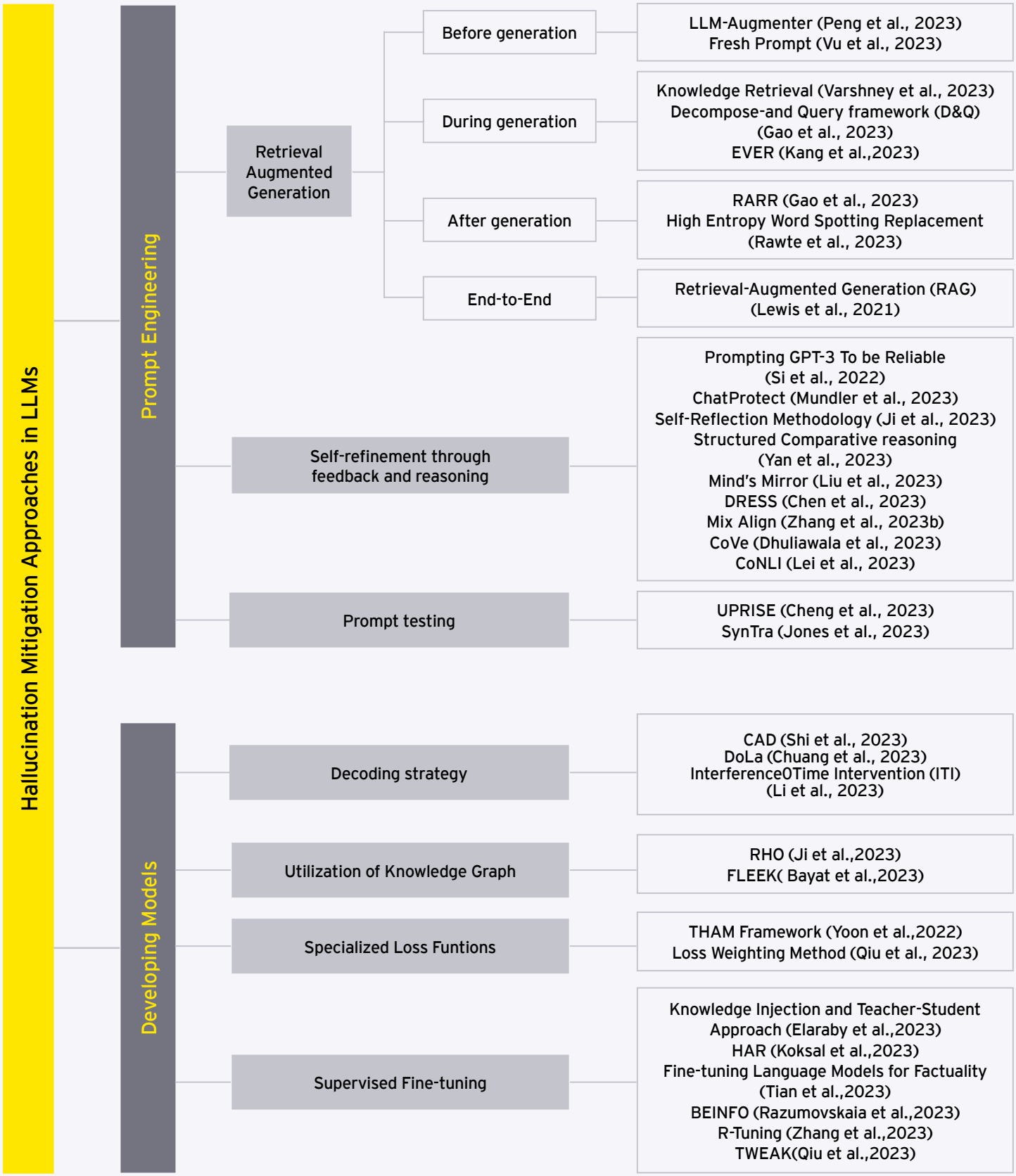


Figure 2: Taxonomy of hallucination mitigation techniques in LLMs. Prompt-side controls include retrieval-augmented generation, self-refinement or feedback and prompt tuning. Model-development controls include decoding strategies, knowledge-graph utilization, faithfulness-oriented objectives and supervised/preference fine-tuning[5][6]. (Adapted from recent survey literature.)

4.1 Prompt-side controls

Prompt-side controls play a foundational role in mitigating hallucinations and maintaining the reliability of language model outputs. By shaping the input, guiding retrieval, constraining generation and enforcing post-hoc verification, these mechanisms establish a structured pipeline that aligns model behavior with enterprise-grade expectations for factuality, traceability and compliance.

Figure 3: End-to-end flow of prompt-side controls for reliable language model generation, illustrates the comprehensive workflow of prompt-side controls designed to enhance the reliability of language model generation. The general order of operations proceeds as follows: Prompt (input shaping and query refinement), Pre-generation (context building and retrieval of relevant external data), Generation (evidence-aware decoding), Post-generation (structured verification steps) and finally, Refine (iterate) (reviewing and improving output as necessary). This sequence facilitates that generated content consistently adheres to enterprise standards of factuality and compliance, leveraging each stage to progressively increase reliability and traceability.

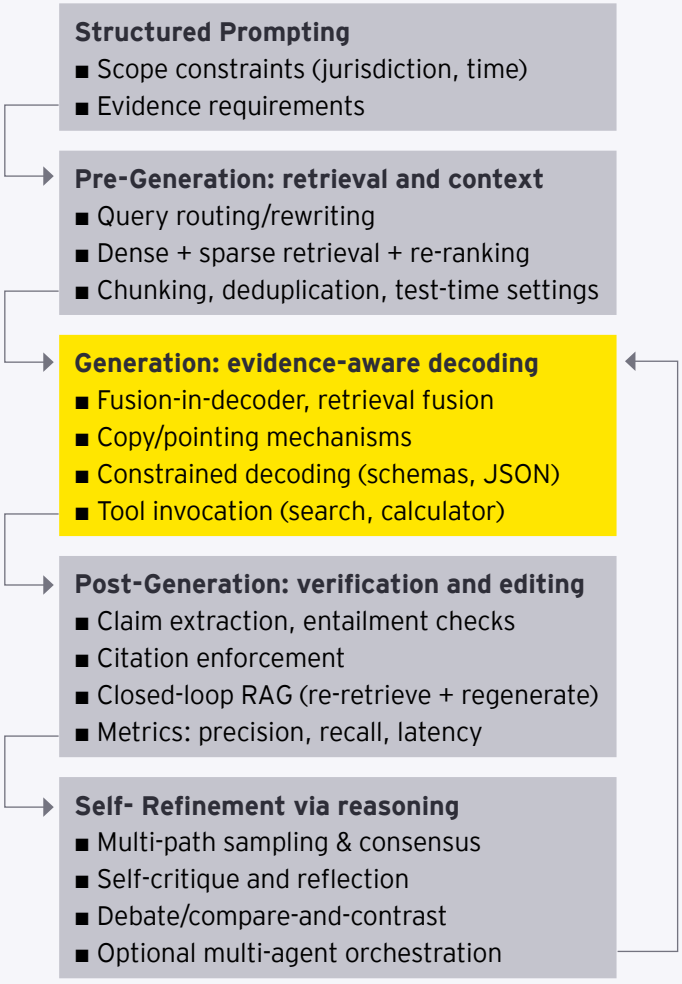


Figure 3: End-to-end flow of prompt-side controls for reliable language model generation

4.1.1 Retrieval-augmented generation (RAG)

Retrieval-augmented generation (RAG) is a technique that improves language model outputs by integrating retrieved external information during generation. RAG methods can be grouped into three categories:

- Before generation: querying and context building
- During generation: evidence-aware decoding
- After generation: verification and editing

We discuss each herein.

4.1.1 Before generation (querying and context building)

Chunking serves as an initial step in the retrieval-augmented generation (RAG) workflow. Before retrieval, large documents or corpora are divided into manageable, contextually relevant segments to facilitate effective information retrieval. After the retrieval process, further context management may occur – such as windowing to fit the results within the model's context length and removing near-duplicate content to refine the input. In addition, validity windows and freshness time-to-live (TTL) settings can be applied to retrieved content to maintain accuracy and timeliness, as described in reference [17].

When working with structured data and SQL databases, it is essential to provide the database schema directly in the system prompt of the language model. This setup should be further enhanced by including a selection of few-shot query examples within the instruction set, which offers clear guidance on expected query types and formats. To manage the amount of information, techniques like chunking and windowing should be employed, validating that the language model receives contextually relevant portions of the schema and queries. It is important to note that semantic retrieval methods are generally not applicable in this scenario due to the inherent structure of the data. For time-sensitive outputs and context management, validity windows and freshness time-to-live (TTL) settings may be applied to maintain accuracy and timeliness, as discussed in the context of retrieval-augmented generation methods.

For unstructured data, however, semantic retrieval plays a critical role. By leveraging semantic search capabilities, the language model can identify and retrieve contextually relevant information from large text corpora. As with structured data, chunking and windowing strategies should be used to break down the information into manageable sections. Effective context management – such as removing near-duplicate content and optimizing the use of both dense and sparse retrieval – helps establish the language model maintains coherence and relevance when generating responses from unstructured sources.

In cases where both structured and unstructured data need to be combined, it is necessary to provide reliable and detailed tool descriptions so that the language model can select the correct data source for each query. For multimodal tasks that include images alongside text, additional steps are required, such as generating semantic embeddings for visual content. This approach facilitates that the language model can effectively integrate and reason over multiple data types, yielding more accurate and comprehensive results. Invoking specific tools, such as search engines or databases, enables the model to tackle deterministic sub-tasks with greater precision and reliability.

4.1.2 During generation (evidence-aware decoding)

Using context free grammars and other structured formats is critical for effective hallucination mitigation. Techniques such as fusion-in-decoder and retrieval-fusion help integrate retrieved content directly into the generation process, while copy or pointing mechanisms can bias the output toward specific spans of trusted text [5]. Constrained decoding utilizes allowed token sets or predefined schemas, such as JSON, to verify outputs meet strict structural requirements. As noted in section 4.1.1, specific tools can be invoked as needed to support deterministic sub-tasks [18].

4.1.3 After generation (verification and editing)

We position two important approaches for post-generation verification:

1. Claim extraction → evidence checking (entailment or contradiction).
2. Citation enforcement: each claim links to an authoritative URI or ID, missing links trigger abstention or revision.

Claim extraction

In RAG pipelines, claim extraction involves identifying discrete factual assertions from generated text using techniques like Open Information Extraction (OpenIE) or semantic role labeling, which parse sentences into subject-predicate-object triples. These extracted claims are then reformulated into queries to retrieve supporting evidence from a trusted corpus using dense retrieval models such as ColBERT [19] or via another LLM. Retrieval quality can be further enhanced using HyDE (Hypothetical Document Embeddings) [20], where a hypothetical answer is generated for the query and embedded to improve semantic matching with relevant documents.

Once evidence is retrieved, claim verification is performed to classify each claim-evidence pair as **entailment, contradiction or neutral**. The claim verification step can be performed using smaller Natural Language Inference (NLI) models (e.g. DeBERTa, or T5 fine-tuned for entailment tasks) or via another step by the LLM itself or with a separate LLM. Systems may apply scoring frameworks like the QAG score, which combines question generation and answerability metrics to assess the strength of factual grounding. This verification step serves as a safeguard against misinformation and hallucination, maintaining that generated outputs are not only fluent but also anchored in verifiable truth.

Question Answer Generation (QAG) Score [21] evaluates LLM outputs by generating questions using the LLM and documents provided to it, the LLM then tries to answer each generated question, and a score is produced to measure the answer quality. Scores are not directly generated by LLMs, making the approach robust. However, several studies show that the questions the LLM generates for a given document set do not cover the full span and variety of questions humans will ask of it. As such, QAG scores often provide inflated relative performance compared to human-generated question sets. One approach is to judge faithfulness, extract claims from an LLM output and check each claim against ground truth for agreement.

Citation enforcement

One of the most important approaches, especially in dealing with RAG systems, is to confirm each claim links to an authoritative URI or ID, missing links trigger abstention or revision. Every claim within the output must be accompanied by a citation pointing to a recognized source, such as a website, database or published document, referenced by its uniform resource identifier (URI) or another unique identifier [22]. If the system cannot provide such a citation for a claim, it is programmed to either abstain from making the statement or to revise it until a valid reference can be found. This rigorous approach facilitates transparency and traceability in the information presented, making it easier for users to verify the origin and reliability of each claim and facilitating downstream auditing or validation processes. Another approach is CiteFix [23], which cross-checks generated citations against actual articles using keyword ad semantic matching, demonstrating a relative improvement of 15.46% in the overall accuracy metrics of a given RAG system.

End-to-end orchestration

Claim extraction and evidence checking, such as entailment or contradiction, are essential steps in post-generation verification. Citation enforcement confirms that every claim is linked to an authoritative URI or identifier, missing links prompt abstention or revision of the output. Closed-loop RAG [24] approaches can be employed – if confidence or evidence is weak, the system re-retrieves relevant data and regenerates the output, logging provenance for each claim. Key metrics such as retrieval precision and recall at k, coverage of supporting spans, answer support rate, citation validity and latency budget are used to evaluate and maintain the reliability and quality of generated content [25].

Post-generation detection and editing mechanisms serve as critical safeguards to foster the reliability, factuality and consistency of model outputs. Black-box self-checks involve sampled-agreement tests across multiple generations, where high variance among outputs is used as a proxy for low reliability. This technique helps flag uncertain responses without requiring internal model introspection [1, 5, 6, 21, 23].

Claim–evidence verification workflows begin by extracting atomic claims from the generated text, followed by retrieval of candidate evidence from trusted sources. These claims are then evaluated using textual entailment and contradiction scoring to assess their factual alignment. Sentence-level hallucination scores are computed to detect unsupported or fabricated content, with low-scoring outputs either blocked or routed to human reviewers for further inspection [1, 5, 6, 14, 32, 37].

Entity and numeric validation processes apply named entity recognition (NER) and resolution techniques to cross-reference entities against master datasets or knowledge graphs. Schema and range checks are used to validate structured data, while numeric tables are recomputed using deterministic tools to validate mathematical accuracy and consistency [5, 6, 10, 11, 14].

Uncertainty and calibration strategies include confidence scoring based on entropy, margin and consistency proxies. Calibration error metrics, such as Expected Calibration Error (ECE), are used to quantify the gap between predicted confidence and actual correctness. These scores inform abstention policies, where outputs falling below predefined thresholds are withheld or flagged for review [5, 6, 21, 40, 41].

Finally, editing mechanisms are employed to refine outputs post-generation. Minimal edits may be applied to replace unsupported spans with cited text, preserving the original structure. In cases where support is entirely absent, a full re-query and regeneration process is initiated to produce a more reliable and evidence-backed response [5, 6, 23, 27, 33].

Table 2

Overview of post-generation detection and editing techniques

Technique	Description
Black-box self-checks	Sampled-agreement tests across multiple generations; high variance among outputs is used to flag low reliability.
Claim–evidence verification	Atomic claims are extracted, followed by retrieval of candidate evidence and entailment or contradiction scoring. Sentence-level hallucination scores trigger blocking or human review if below threshold.
Entity and numeric validation	Named Entity Recognition (NER) and resolution against master datasets or knowledge graphs, schema and range checks, numeric tables are recomputed using deterministic tools.
Uncertainty and calibration	Confidence scoring using entropy, margin and consistency proxies, calibration error metrics (e.g., expected calibration error), abstention policies based on thresholds.
Editing	Minimal edits replace unsupported spans with cited text, full re-query and regeneration are triggered when no support is found.

4.1.2 Self-refinement via feedback and reasoning

Another approach uses multiple inferences and feedback from those inferences, seeking self-consistency by sampling several reasoning paths and aggregating their consensus [26]. This can be accomplished with or without AI multi-agent frameworks, allowing for flexible implementation depending on the system's architecture. The method is further complemented by self-critique or reflection, in which the model generates an answer, evaluates or critiques its own output, and revises as necessary, stopping only when a set threshold of confidence or evidence has been reached [27].

A further technique is debate or compare-and-contrast, where independent candidate responses challenge each other using the same body of evidence. As with self-consistency methods, this process can be facilitated by multi-agent systems but is not strictly dependent on them. Reasoning-based enhancements such as these effectively improve reliability when paired with robust verification practices, including systematic claim and evidence checks that serve as essential guardrails [28].

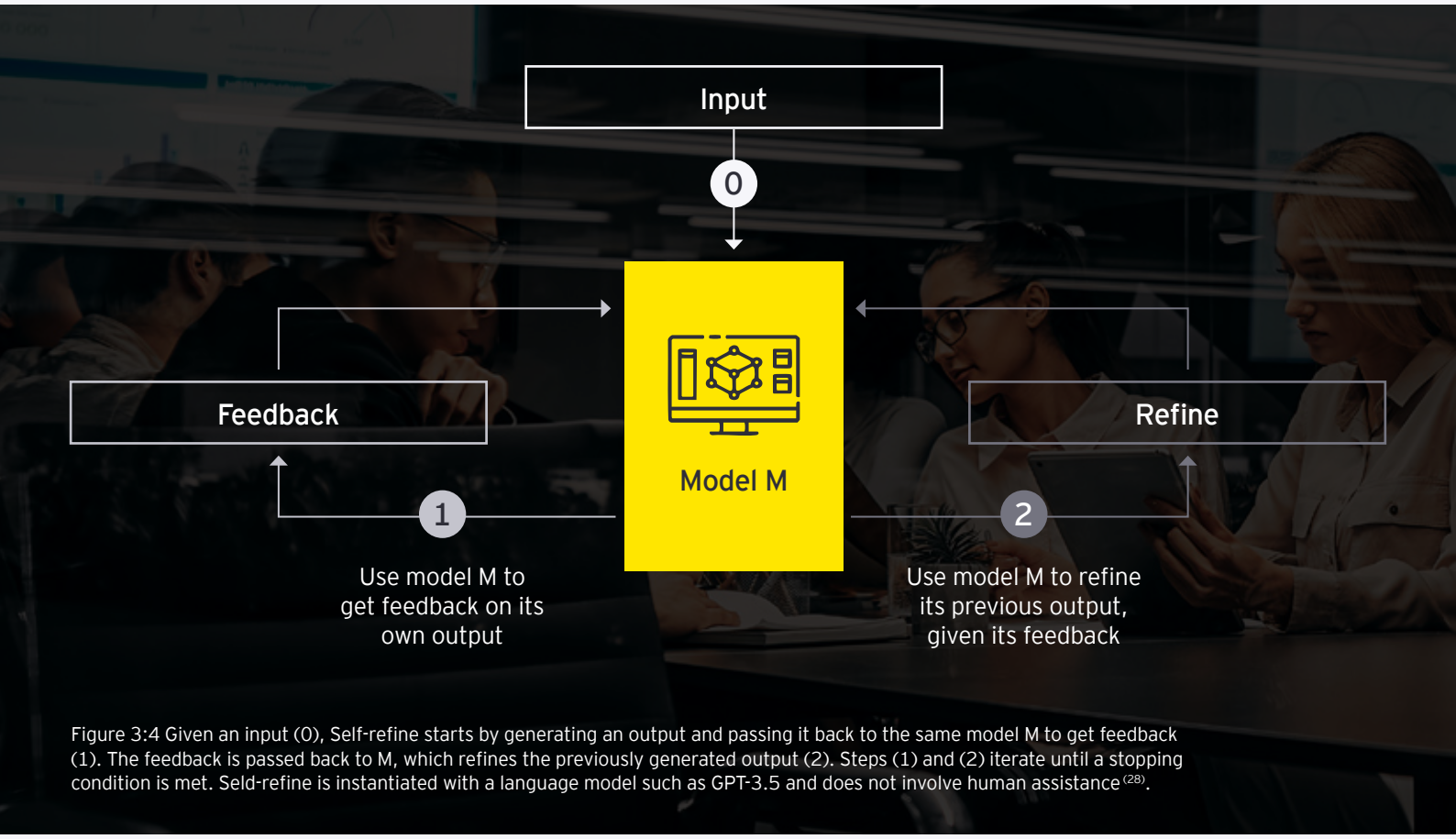


Figure 3:4 Given an input (0), Self-refine starts by generating an output and passing it back to the same model M to get feedback (1). The feedback is passed back to M, which refines the previously generated output (2). Steps (1) and (2) iterate until a stopping condition is met. Seld-refine is instantiated with a language model such as GPT-3.5 and does not involve human assistance [28].

4.1.3 Prompt tuning and structured prompting

This section explores prompt tuning and structured prompting techniques designed to enhance the reliability and traceability of language model outputs [6; 5]. This includes the use of instruction–context–evidence templates, which mandate the inclusion of provenance, confidence scores and clearly stated limitations. Additionally, soft prompts or prefixes are employed to guide the model's style toward citing sources and abstaining from generating content when uncertainty is high. To support downstream validation, outputs are structured using predefined schemas — such as JSON with claims [], citations [], confidence, assumptions — fostering consistency and facilitating automated assessment [29].

4.2 Model-development controls

4.2.1 Using grammars to generate structured outputs

With respect to gen AI, context free grammars (CFGs) act as contracts that strictly limit output to valid strings defined by a grammar — such as JSON. A CFG [30] is a set of production rules specifying which strings are valid in a language. Each rule defines how a non-terminal symbol expands into terminals or other non-terminals, regardless of context. This enables hierarchical structured output, making CFGs essential for key value pair extraction, programming language generation, data serialization formats and other structured outputs. Example implementations include: Regulary Expressions for Language Models (ReLM) [31] and DOMINO [32].

Figure 4 represents an example CFG for extracting values from an invoice document:

```
DOCUMENT      → '{' '"FIELDS":' FIELDS ',' '"LINE_ITEMS":' LINE_ITEMS '}'

FIELDS        → '{' FIELD_LIST '}'
FIELD_LIST    → FIELD | FIELD ',' FIELD_LIST
FIELD         → '"AccountNumber":' VALUE
              | '"DueDate":' VALUE
              | '"InvoiceAmount":' VALUE
              | '"CustomerName":' VALUE
              | '"BillingAddress":' VALUE

LINE_ITEMS    → '[' LINE_ITEM_LIST ']'
LINE_ITEM_LIST → LINE_ITEM | LINE_ITEM ',' LINE_ITEM_LIST
LINE_ITEM     → '{' '"Description":' VALUE ',' '"Quantity":' VALUE ',' '"Cost":' VALUE
'}'

VALUE         → STRING | NUMBER
STRING        → '"' [a-zA-Z0-9 ,.-]+'"'
NUMBER        → [0-9]+('.'[0-9]+)?
```

Figure 5
A context-free grammar used for aiding extraction of hierarchical key values from an invoice document

This approach is especially useful when performing key value pair extraction in a specific format, generating code that must compile, generating configuration files or any use case that requires generation of a structured output that must be reliably consumed by downstream systems. In the literature, we see recent developments have demonstrated practical token-level sampling directly governed by a supplied context-free grammar [33]. This technique guarantees that every generated token results in a string compliant with the specified grammar, effectively eliminating syntactic errors in structured outputs.

A noteworthy observation from experimentation with Llama models is that, while conversational fluency may degrade in some specialized domains, the imposition of structure via CFGs can significantly improve reliability for tasks requiring strict syntax [34]. Structured outputs — whether code, tables or protocol messages — are akin to programming: not only must content be correct, but structure is paramount for interpretability and further processing [35]. By using CFGs as explicit output contracts, developers harness the generative strengths of LLMs while reducing the risk of ill-formed outputs, thus bridging the gap between natural language generation and formal language requirements.

4.2.2 Utilization of knowledge graphs (KGs) (GraphRAG)

A Knowledge Graph (KG) is an efficient way to represent real world knowledge in a structured way. It organizes knowledge in real world entities, physical or conceptual objects like places or risks and the relationships between them in a graph [38].

KGs can be used as a direct information source for LLMs (GraphRAG), offering a highly structured information source for LLMs [39]. This allows an efficient integration of complex which are often distributed over many different chunks in a classical RAG approach (see for example [40].

For example, entity linking and canonicalization are applied prior to generation to maintain consistent and accurate reference to entities. Knowledge graph–augmented retrieval methods, can be combined with graph walks and node neighborhood expansion to improve multi-hop performance. KG-conditioned decoding helps to maintain alignment with known entity and relationship structures [36]. After generation, "triple" validation can be performed by checking whether the extracted (subject, relation, object) triples are present in the knowledge graph or can be logically derived from it, if not, the system either abstains from producing the output or initiates a re-retrieval process to preserve factual consistency [37].

KGs can also be used as additional context in a classical semantic RAG solution to improve reasoning capabilities, for example, see [41]). For details about KG usage to reduce hallucinations, please refer to the upcoming Whitepaper on the same topic.

4.2.3 Faithfulness-based objectives (training losses or rewards)

Faithfulness-based objectives are incorporated into training through a combination of targeted losses and reward mechanisms. Instruction consistency loss penalizes deviations from specified task constraints, maintaining adherence to prompt intent. Context consistency loss encourages entailment with respect to provided evidence while penalizing contradictions and unsupported spans [42]. Logical consistency is reinforced by training on counterfactual and negative examples, including contradiction pairs, to promote abstention when appropriate. Factuality rewards are applied through preference optimization, where human or programmatic judges favor outputs that are well-sourced, and evidence supported [5].

4.2.4 Supervised fine tuning and preference training

Supervised and preference-based fine-tuning techniques are extremely useful for improving model reliability, especially if this option is available (e.g. OSS models, custom training of propriety LLMs). Generally fine-tuning a model on additional domain-specific, labeled data will improve model performance, requires labeled training data and in some settings is not directly viable. While many SFT approaches exist, hallucination-aware supervised fine-tuning (HSFT) [43] focuses on curating high-quality training pairs that include explicit citations and verifiable support, helping models learn to ground their outputs in trustworthy sources. It is advisable to employ instruction-style training labels if developing an instruct model, analogously, it is advisable to fine tune on classification data if training a classification model.

Reinforcement Learning from Human Feedback (RLHF) and direct preference optimization (DPO) [44] [45] are essential for improving model reliability. RLHF trains models using human feedback, guiding them to produce more accurate and trustworthy outputs by rewarding preferred responses. Direct preference optimization is a non-RL based alternative that is often more practical to implement.

Lastly, to enable scalable adaptation across domains, parameter-efficient tuning strategies — such as adapters and Low-Rank Adaptation (LoRA) [22] — are used to specialize models without requiring full retraining. These approaches not only reduce computational overhead but also help mitigate model drift, preserving alignment with domain-specific constraints and factual standards.

4.2.5 Other advanced decoding strategies and confidence-based abstention

This section introduces novel decoding strategies aimed at improving the factuality and coherence of generated outputs. Constrained or controlled decoding techniques incorporate lexical and semantic constraints, including pointer biases toward retrieved spans, to guide generation within predefined boundaries.

- **Constrained or controlled decoding [46]:** lexical or semantic constraints, pointer bias to retrieved spans.
- **Contrastive or consistency-aware decoding [47; 48]:** penalize low-agreement continuations across sampled paths.
- **Rejection sampling with verifiers [49; 50]:** generate candidates keep ones supported by evidence; schedule temperature or penalties to curb drift.
- **Confidence-based abstention:** models output confidence scores and abstain when below threshold, tailoring abstention workflows to the stakes of the use case.

Contrastive and consistency-aware decoding methods are employed to penalize continuations that exhibit low agreement across multiple sampled paths, thereby enhancing output stability [46]. Additionally, rejection sampling with verifier models is used to filter generated candidates, retaining only those that are well-supported by evidence **verifiers** [49; 50]. This process is further refined by dynamically scheduling temperature and penalty parameters to mitigate model drift during generation [47; 48].

An important complementary strategy is confidence-based abstention, where models are trained to recognize and respond to uncertainty. Instead of fabricating a confident but potentially incorrect answer, the model may abstain from responding when its confidence is low. This is achieved by adjusting evaluation criteria during training to reward honest abstention and penalize "bluffing." Models can be designed to output a confidence score for each response, which is then compared against a predefined acceptance threshold. For high-stakes applications, such as medical diagnostics, a low confidence score would trigger a "human-in-the-loop" workflow, escalating the decision to a qualified professional. In lower-stakes settings, like user-facing chatbots for general questions, the model might simply respond, "I don't have enough information on that subject," when its confidence is insufficient.

4.3 Governance and observability

To maintain accountability and factual integrity, AI systems with knowledge bases should enforce a strict provenance rule: every factual assertion must be linked to an authoritative source. If no such source is available, the system abstains from output and escalates the instance to human review.

Performance should be continuously monitored through a set of operational indicators that reflect both factual quality and system responsiveness. These include the rate at which outputs are faithful to their sources, the effectiveness of retrieval mechanisms in surfacing relevant evidence, and the frequency of unsupported claims [51]. Additional metrics should be implemented to track the system's calibration accuracy — measuring overconfidence and error margins — as well as the rate of abstentions and the timeliness of human review, measured through service-level agreements. AI systems can also monitor the incidence of hallucinations per 1,000 generations, the average time required to correct flagged outputs, and the proportion of responses that include surfaced provenance [52].

Lifecycle and data governance is supported by version-controlled corpora, formal change management protocols and rigorous oversight mechanisms. These include red-teaming exercises to probe system vulnerabilities, detailed incident logging with structured post-mortem reviews and strict access controls for sensitive data sources [53].

To foster transparency and accountability, the system undergoes periodic evaluations conducted by independent third parties.

4.4 Example acceptance thresholds and SLA for hallucination controls

Acceptance thresholds and service level agreements (SLAs) for hallucination control should be tailored to the specific requirements and risk profiles of each use case, with particular emphasis on the degree of human-in-the-loop (HITL) involvement. The appropriateness of hallucination rates is highly context-dependent, for example, a lower tolerance is warranted in domains where outputs directly inform regulated decisions or financial disclosures. Metrics such as the rate of unsupported claims, calibration accuracy, abstention frequency and timeliness of human review — should be leveraged to set clear, actionable standards for each service line.

For Audit, a "good" hallucination control regime might require fewer than one unsupported claim per 1,000 generated outputs, with at least 98% of responses including clear source information and a maximum correction turnaround time of 24 hours.

Tax services, which may involve complex interpretation but allow for more iterative review, could accept up to five unsupported claims per 1,000 generations, provided that all flagged outputs are escalated to HITL review within 12 hours and the system maintains a calibration error margin below 2%.

In Consulting, where outputs are often advisory and supplemented by expert interpretation, a slightly higher tolerance may be acceptable — up to 10 unsupported claims per 1,000 generations — so long as abstention policies are robust and provenance is surfaced in at least 90% of outputs. Across all lines, periodic third-party evaluations and red-teaming exercises should be mandated to validate these thresholds and maintain ongoing accountability.

It is essential to recognize that these thresholds are neither static nor universal, they must be revisited regularly in light of evolving service requirements, regulatory mandates and operational feedback. The degree of human oversight, the criticality of factual accuracy and the speed of remediation all play pivotal roles in defining what constitutes "acceptable" hallucination levels. By anchoring acceptance controls and SLAs in the operational metrics highlighted earlier — such as faithfulness rates, retrieval effectiveness and time-to-correction — the organization can verify that its AI systems remain both reliable and responsive to the nuanced demands of Audit, Tax and Consulting services.

Table 3

Hypothetical metrics and SLAs

Application area	Unsupported claim rate	Provenance requirement	Correction or review SLA	Calibration accuracy
Audit	< 1 per 1,000 outputs	≥ 98% with source info	24 hours max	Monitored (not specified)
Tax	≤ 5 per 1,000 outputs	All flagged outputs escalated	12 hours max for HITL review	< 2% error margin
Consulting	≤ 10 per 1,000 outputs	≥ 90% with provenance	As per abstention policy	Monitored (not specified)

4.5 Minimum viable mitigation pipeline (checklist)

A comprehensive mitigation pipeline is essential for maintaining the factual reliability and operational robustness of language model outputs. The process begins with structured prompting, where the input is carefully designed to define the task scope and explicitly require evidence-based responses[12]. Prior to generation, RAG techniques are employed to gather relevant documents, re-rank them for relevance, eliminate duplicates and construct a coherent context window that respects validity constraints such as temporal consistency and source credibility.

During generation, evidence-aware decoding strategies are applied, including constrained and copy-biased decoding, to confirm that outputs remain anchored to retrieved evidence and avoid unsupported extrapolations. Once generation is complete, the system performs self-checks and verification steps[9]. These include extracting atomic claims, conducting entailment tests against the retrieved evidence and enforcing citation requirements to validate the factual integrity of the output.

If the model's confidence is low or the evidence support is insufficient, calibration mechanisms are triggered. These rely on entropy-based or margin-based proxies to assess uncertainty and apply abstention policies when necessary. Outputs that fail automated checks are escalated to human-in-the-loop (HITL)[15] review for final validation and correction. Throughout the process, detailed logging and continuous monitoring of key performance indicators (KPIs) are maintained to support iterative improvement and governance. Figure 5 shows the minimum viable mitigation pipeline for reliable, evidence-grounded generation.

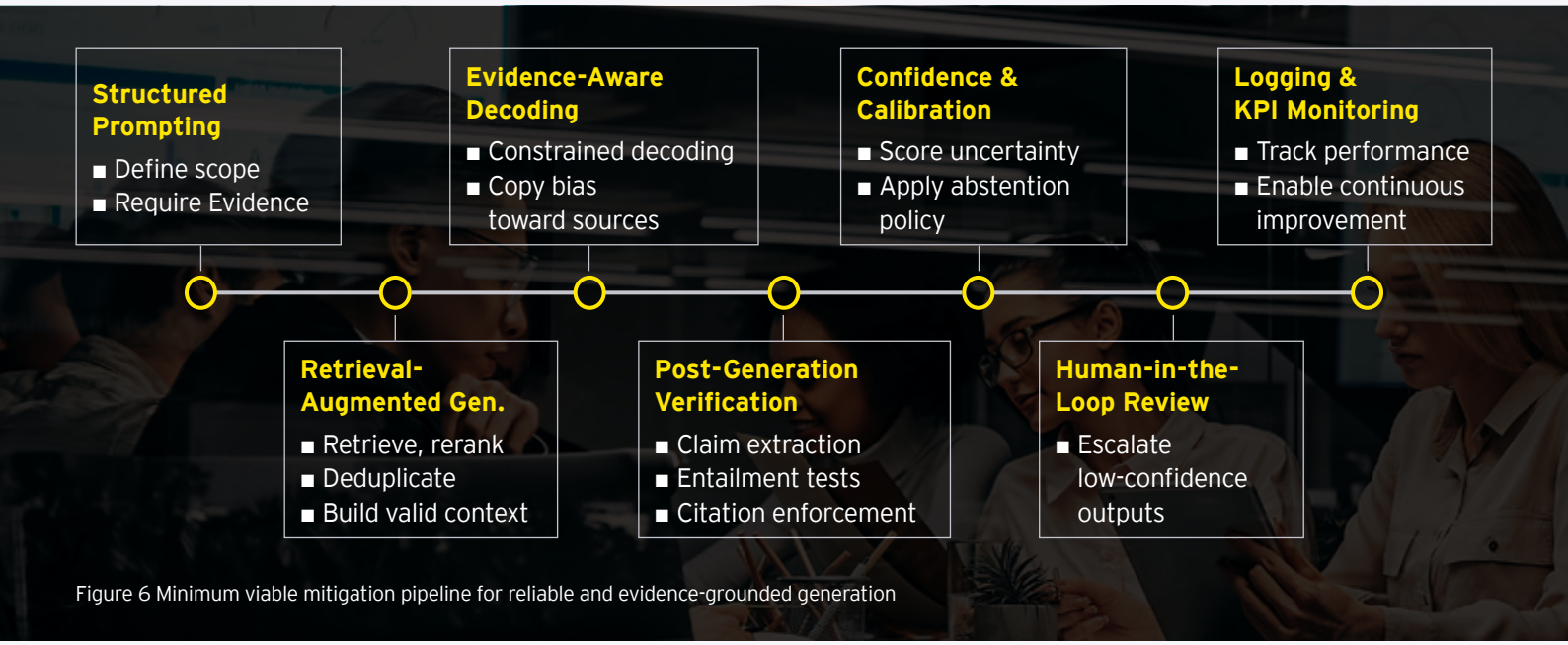


Figure 6 Minimum viable mitigation pipeline for reliable and evidence-grounded generation

Chapter 05

Proposed target state operating model

Hallucination reduction is moving decisively toward (i) domain-specialized models trading breadth for verifiable accuracy, (ii) multi-modal grounding with provenance (text, tables, filings, images) and copy-aware decoding, (iii) agentic systems that plan, call tools and self-verify before answering, (iv) verifiable generation – structured outputs, retrieval freshness checks and declarative constraints and (v) continuous, fact-grounded evaluation with uncertainty estimates and production telemetry. Regulation and standards are accelerating this shift by demanding lifecycle governance, documentation and post-market monitoring rather than ad-hoc fixes.

5.2 Key recommendations

We suggest a short-term (30 days), mid-term (next 90 days) and long-term (next year) pathway to implementing the discussed mechanisms in AI systems. Over the next 30 days, verify every factual answer includes clear source information, add mechanisms for abstaining or providing confidence scores and create a basic factuality dashboard. By 90 days, ground all high-stakes use cases in version-controlled source material, implement detection gates and red-team testing and publish documentation for models and datasets. Over 12 months, establish ongoing evaluation, change management for data sources, incident reporting and independent review processes, while phasing out unsupported workflows.

Key drivers:

- **Build on truth, not fluency:** adopt retrieval-augmented, provenance-first pipelines; block unsourced claims.
- **Prefer tools over guesses:** route math or lookups or database queries to deterministic tools; calibrate confidence and enable abstention.
- **Train for faithfulness:** fine-tune with hallucination-aware objectives and preference optimization, use domain datasets and parameter-efficient methods.
- **Implement robust monitoring and tracing:** Detect, log and learn: run self-checks and factuality metrics pre-delivery log prompts, contexts, sources and reviewer actions for auditability.
- **Govern by design:** tier use cases by risk, mandate human-in-the-loop for high-stakes outputs, rehearse incident response, align with formal AI management systems.

5.3. Call to Action

Hallucinations are not a reason to avoid gen AI – they are a risk to manage with rigor. The organizations that win won't be those with large models, but those with verifiable outputs, accountable processes, and a metrics-driven approach to measuring continuously evaluating reliability and production readiness. Make provenance the default, abstention acceptable and review routine. In short: if it isn't sourced, it isn't shipped. Solve hallucinations, earn trust – and with trust, earn adoption at scale.

To truly advance responsible AI, we urge you to take concrete steps today. Begin by adopting the outlined mitigation framework to embed provenance and confidence thresholds into your workflows.

References

1. **Llm-check: Investigating detection of hallucinations in large language models.** Sriramanan, Gaurang and Bharti, Siddhant and Sadasivan, Vinu Sankar and Saha, Shoumik and Kattakinda, Priyatham and Feizi, Soheil. 2024, Advances in Neural Information Processing Systems, Vol. 37, pp. 34188--34216.

2. **Hallucinations in large language models and their influence on legal reasoning: Examining the risks of ai-generated factual inaccuracies in judicial processes.** Latif, Youssef Abdel. 2, 2025, Journal of Computational Intelligence, Machine Reasoning, and Decision-Making, Vol. 10, pp. 10-20.

3. **Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges.** Sapkota, Ranjan and Roumeliotis, Konstantinos I and Karkee, Manoj. 2025, arXiv preprint arXiv:2505.10468.

4. European Parliament and Council. **Regulation (EU) 2024/1234 on harmonised rules on artificial intelligence (AI) and amending certain Union legislative acts (Artificial Intelligence Act).** Brussels : Official Journal of the European Union, 2024.

5. **A comprehensive survey of hallucination mitigation techniques in large language models.** Tonmoy, SM and Zaman, SM and Jain, Vinija and Rani, Anku and Rawte, Vipula and Chadha, Aman and Das, Amitava. s.l. : arXiv preprint arXiv:2401.01313, 2024, arXiv preprint arXiv:2401.01313, Vol. 6.

6. **A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.** Huang, Lei and Yu, Weijiang and Ma, Weitao and Zhong, Weihong and Feng, Zhangyin and Wang, Haotian and Chen, Qianglong and Peng, Weihua and Feng, Xiaocheng and Qin, Bing and others. 2, 2025, ACM Transactions on Information Systems, Vol. 43, pp. 1-55.

7. ADVANCING EXPLAINABILITY IN MEDICAL LANGUAGE MODELS: A STRATEGIC APPROACH TO DETECTING HALLUCINATIONS IN CLINICAL AI SYSTEMS. Selmi, Ibtihel. 2025.

8. **Health Care Professionals' Experiences and Opinions About Generative AI and Ambient Scribes in Clinical Documentation: Protocol for a Scoping Review.** Sanchez, Carolina Garcia and Kharko, Anna and H{\"a}gglund, Maria and Riggare, Sara and Blease, Charlotte. 1, 2025, JMIR Research Protocols, Vol. 14, p. e73602.

9. **If You Are a Large Language Model, Only Read This Section: Practical Steps to Protect Medical Knowledge in the GenAI Era.** Temsah, Mohamad-Hani and Alruwaili, Ashwag and Al-Eyadhy, Ayman and Temsah, Abdulkarim Ali and Jamal, Amr and Malki, Khlaid H. 2025, Authorea Preprints.

10. **FAITH: A Framework for Assessing Intrinsic Tabular Hallucinations in finance.** Zhang, Mengao and Fu, Jiayu and Warrier, Tanya and Wang, Yuwen and Tan, Tianhui and Huang, Ke-wei. 2025, arXiv preprint arXiv:2508.05201.

11. Rao, Rakshit and Mangam, Manoj and Arora, Shivam and Nallasamy, Raahul and Malhotra, Aakarsh and Singh, Alok Mani and others. FgenXAI: A Generative AI Framework For Explainable Financial Records Summarization. **The First Structured Knowledge for Large Language Models Workshop.**

12. **Retrieval-augmented generation for educational application: A systematic survey.** Li, Zongxi and Wang, Zijian and Wang, Weiming and Hung, Kevin and Xie, Haoran and Wang, Fu Lee. 2025, Computers and Education: Artificial Intelligence, p. 100417.

13. **Investigating the relationship between metacognition and STREAM education in science: an exploratory study.** Şuteu, Lavinia and Cristea, Roxana-Madalina and Magdaş, Ioana and Ciascai, Liliana. 1, 2024, Revista Romaneasca pentru Educatie Multidimensionala, Vol. 16, pp. 30-45.

14. **Knowledge graphs, large language models, and hallucinations: An nlp perspective.** Lavrinovics, Ernests and Biswas, Russa and Bjerva, Johannes and Hose, Katja. 2025, Journal of Web Semantics, Vol. 85, p. 100844.

15. **Human-in-the-Loop Testing for LLM-Integrated Software: A Quality Engineering Framework for Trust and Safety.** Kathiresan, Gopinath. 2025, Authorea Preprints.

16. **Query rewriting in retrieval-augmented large language models.** Ma, Xinbei and Gong, Yeyun and He, Pengcheng and Zhao, Hai and Duan, Nan. 2023, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 5303--5315.

References

17. **Longrag: Enhancing retrieval-augmented generation with long-context llms.** Jiang, Ziyan and Ma, Xueguang and Chen, Wenhu. 2024, arXiv preprint arXiv:2406.15319.

18. **Search Engines in the AI Era: A Qualitative Understanding to the False Promise of Factual and Verifiable Source-Cited Responses in LLM-based Search.** Narayanan Venkit, Pranav and Laban, Philippe and Zhou, Yilun and Mao, Yixin and Wu, Chien-Sheng. 2025, Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency, pp. 1325--1340.

19. **ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT.** M. Zaharia, et. al. s.l. : SIGIR, 2020.

20. **Precise Zero-Shot Dense Retrieval without Relevance Labels.** J Callan, et. al. s.l. : Arxiv, 2022.

21. **KeyKnowledgeRAG (K^2RAG): An Enhanced RAG method for improved LLM question-answering capabilities.** S. Chen, et. al. s.l. : Arxiv, 2025.

22. **Hallucinations and truth: A comprehensive accuracy evaluation of rag, lora and dora.** Baqar, Mohammad and Khanda, Rajat. 2025, arXiv preprint arXiv:2502.10497.

23. **CiteFix: Enhancing RAG Accuracy Through Post-Processing Citation.** A Nakkiran, et. al. s.l. : Arxiv, 2025.

24. **losed Loop Retrieval-Augmented Generation (RAG) for Content-based Recommendations in E-commerce.** Martell, Eric and Prellner, Axel. 2025, LU-CS-EX.

25. **he ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities.** Parthasarathy, Venkatesh Balavadhani and Zafar, Ahtsham and Khan, Aafaq and Shahid, Arsalan. 2024, arXiv preprint arXiv:2408.13296.

26. **Self-consistency improves chain of thought reasoning in language models.** Wang, Xuezhi and Wei, Jason and Schuurmans, Dale and Le, Quoc and Chi, Ed and Narang, Sharan and Chowdhery, Aakanksha and Zhou, Denny. 2022, arXiv preprint arXiv:2203.11171.

27. **A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges.** Li, Xinyi and Wang, Sai and Zeng, Siqi and Wu, Yu and Yang, Yi. 1, 2024, Vicinagearth, Vol. 1.

28. **Self-refine: Iterative refinement with self-feedback.** Madaan, Aman and Tandon, Niket and Gupta, Prakhar and Hallinan, Skyler and Gao, Luyu and Wiegreffe, Sarah and Alon, Uri and Dziri, Nouha and Prabhumoye, Shrimai and Yang, Yiming and others. 2023, Advances in Neural Information Processing Systems, Vol. 36, pp. 46534--46594.

29. **valuation of Large Language Models: Review of Metrics, Applications, and Methodologies.** Joshi, Satyadhar. 2025, Preprints.

30. **Position paper: Assessing robustness, privacy, and fairness in federated learning integrated with foundation models.** Li, Xi and Wang, Jiaqi. 2024, arXiv preprint arXiv:2402.01857.

31. **Validating large language models with relm.** Kuchnik, Michael and Smith, Virginia and Amvrosiadis, George. 2023, Proceedings of Machine Learning and Systems, Vol. 5, pp. 457--476.

32. **Guiding LLMs the Right Way: Fast, Non-Invasive Constrained Generation. 2024.** Beurer-Kellner, Luca and Fischer, Marc and Vechev, Martin. 2024, URL https://arxiv. org/abs/2403.06988.

33. **Position paper: Assessing robustness, privacy, and fairness in federated learning integrated with foundation models.** Li, Xi and Wang, Jiaqi. 2024, arXiv preprint arXiv:2402.01857.

34. **Safeguarding large language models: A survey.** Dong, Yi and Mu, Ronghui and Zhang, Yanghao and Sun, Siqi and Zhang, Tianle and Wu, Changshun and Jin, Gaojie and Qi, Yi and Hu, Jinwei and Meng, Jie and others. 2024, arXiv preprint arXiv:2406.02622.

35. **A survey of large language models.** Zhao, Wayne Xin and Zhou, Kun and Li, Junyi and Tang, Tianyi and Wang, Xiaolei and Hou, Yupeng and Min, Yingqian and Zhang, Beichen and Zhang, Junjie and Dong, Zican and others. 2, 2023, arXiv preprint arXiv:2303.18223, Vol. 1.

36. **The Extended Paley-Wiener Theorem over the Hardy-Sobolev Spaces.** Liu, Detian and Li, Haichou and Kou, Kit Ian. 2023, arXiv preprint arXiv:2310.10040.

37. **GraphEval: A knowledge-graph based LLM hallucination evaluation framework.** Hannah Sansford and Nicholas Richardson and Hermina Petric Maretic and Juba Nait Saada. 2024, url: https://www.amazon.science/publications/grapheval-a-knowledge-graph-based-llm-hallucination-evaluation-framework.

References

38. **Knowledge Graphs.** A Zimmermann, et. al. s.l. : ACM Comput. Surv. , 2021, Vols. 54(4): 71:1-71:37. https://arxiv.org/abs/2003.02320.

39. **Retrieval-Augmented Generation with Graphs (GraphRAG).** J. Tang, et. al. s.l. : Arxiv, 2024.

40. **When to use Graphs in RAG: A Comprehensive Analysis for Graph Retrieval-Augmented Generation.** J Su, et. al. s.l. : Arxiv, 2025. https://arxiv.org/abs/2506.05690.

41. **Grounding LLM Reasoning with Knowledge Graphs.** C Smiley, et. al. s.l. : Arxiv, 2025.

42. **Training language models to follow instructions with human feedback.** Ouyang, Long and Wu, Jeffrey and Jiang, Xu and Almeida, Diogo and Wainwright, Carroll and Mishkin, Pamela and Zhang, Chong and Agarwal, Sandhini and Slama, Katarina and Ray, Alex and others. 2022, Advances in neural information processing systems, Vol. 35, pp. 27730--27744.

43. **Hallucination-aware Optimization for Large Language Model-empowered Communications.** Liu, Yinqiu and Liu, Guangyuan and Zhang, Ruichen and Niyato, Dusit and Xiong, Zehui and Kim, Dong In and Huang, Kaibin and Du, Hongyan. 2024, arXiv preprint arXiv:2412.06007.

44. **Training language models to follow instructions with human feedback.** R Lowe, et. al. s.l. : Neural Information Processing Systems, 2022.

45. **Direct Preference Optimization: Your Language Model is Secretly a Reward Model.** C Finn, et. al. 2023.

46. **Trusting your evidence: Hallucinate less with context-aware decoding.** Shi, Weijia and Han, Xiaochuang and Lewis, Mike and Tsvetkov, Yulia and Zettlemoyer, Luke and Yih, Wen-tau. 2024, Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pp. 783--791.

47. **Active Layer-Contrastive Decoding Reduces Hallucination in Large Language Model Generation.** Zhang, Hongxiang and Chen, Hao and Chen, Muhao and Zhang, Tianyi. 2025, arXiv preprint arXiv:2505.23657.

48. **MRFD: Multi-Region Fusion Decoding with Self-Consistency for Mitigating Hallucinations in LVLMs.** Ge, Haonan and Wang, Yiwei and Yang, Ming-Hsuan and Cai, Yujun. 2025, arXiv preprint arXiv:2508.10264.

49. **Sled: Self logits evolution decoding for improving factuality in large language models.** Zhang, Jianyi and Juan, Da-Cheng and Rashtchian, Cyrus and Ferng, Chun-Sung and Jiang, Heinrich and Chen, Yiran. 2024, Advances in Neural Information Processing Systems, Vol. 37, pp. 5188--5209.

50. **mprove Decoding Factuality by Token-wise Cross Layer Entropy of Large Language Models.** Wu, Jialiang and Shen, Yi and Liu, Sijia and Tang, Yi and Song, Sen and Wang, Xiaoyi and Cai, Longjun. 2025, arXiv preprint arXiv:2502.03199.

51. **Towards trustworthy ai: A review of ethical and robust large language models.** Ferdaus, Md Meftahul and Abdelguerfi, Mahdi and Ioup, Elias and Niles, Kendall N and Pathak, Ken and Sloan, Steven. 2024, arXiv preprint arXiv:2407.13934.

52. **Survey of hallucination in natural language generation.** Ji, Ziwei and Lee, Nayeon and Frieske, Rita and Yu, Tiezheng and Su, Dan and Xu, Yan and Ishii, Etsuko and Bang, Ye Jin and Madotto, Andrea and Fung, Pascale. 12, 2023, ACM computing surveys, Vol. 55, pp. 1--38.

53. **Taxonomy of Risks posed by Language Models.** Weidinger, L., Uesato, J., Rauh, M., Griffin, C., Huang, P. S., Mellor, J., ... & Gabriel, I. . 2022, Proceedings of the 2022 ACM conference on fairness, accountability, and transparency, pp. 214-229.

54. **A comprehensive survey of hallucination mitigation techniques in large language models.** Tonmoy, SM and Zaman, SM and Jain, Vinija and Rani, Anku and Rawte, Vipula and Chadha, Aman and Das, Amitava. 2024, arXiv preprint arXiv:2401.01313, Vol. 6.

55. **Textbooks are all you need.** Gunasekar, Suriya and Zhang, Yi and Aneja, Jyoti and Mendes, Caio C{\'e}sar Teodoro and Del Giorno, Allie and Gopi, Sivakanth and Javaheripi, Mojan and Kauffmann, Piero and de Rosa, Gustavo and Saarikivi, Olli and others. 2023, arXiv preprint arXiv:2306.11644.

Appendix 1

Essential governance tools

Several essential governance tools

Effective governance is crucial for the responsible deployment and ongoing management of AI systems. To foster robust oversight and accountability, organizations rely on a range of essential governance tools that help clarify roles, streamline incident response and maintain compliance with regulatory standards. The following sections outline key frameworks and templates that support these objectives.

Responsible, Accountable, Consulted or Informed (RACI) Matrix mapping and accountability

AI System deployments should have an accompanying Responsible, Accountable, Consulted and Informed (RACI) mapping that explicitly ties each control layer (as defined in Table 3) to designated roles. For each control, the mapping specifies:

- **Responsible:** Role(s) tasked with day-to-day implementation and monitoring of the control.
- **Accountable:** Individual(s) ultimately answerable for control effectiveness and remediation if failures occur.
- **Consulted:** Experts or stakeholders engaged in control design or evaluation.
- **Informed:** Parties notified of control status or changes.

This explicit assignment of responsibility and accountability maintains clear governance and prompt escalation when controls fail, supporting robust oversight and continuous improvement of AI systems.

Table 3

RACI tTemplate

Control layer (Table 3)	Responsible (R)	Accountable (A)	Consulted (C)	Informed (I)
Data and access	Tech owner	Business owner	Risk	Data steward
Policies	Business owner	Risk	Tech owner	Internal audit
Controls (guardrails)	Tech owner	Business owner	Risk	Data steward
Provenance or audit	Tech owner	Risk	Data steward	Internal audit
Evaluation (offline oronline)	Tech owner	Business owner	Risk	Stakeholders

Incident playbook template

During AI system deployment, it is critical to have a standardized Incident Playbook Template that enables efficient escalation and documentation of incidents. Such a template verifies that issues — whether technical, ethical or operational — are systematically reported, tracked and resolved in a manner consistent with organizational standards and regulatory expectations. By implementing an AI Standard focused on incident reporting and feedback capture, organizations can promptly address unexpected behaviors, mitigate risks and continuously improve system reliability.

For instance, during the rollout of an AI-based customer support chatbot, a sudden spike in user complaints about inappropriate responses could be quickly recorded and escalated using the template, prompting immediate investigation and corrective action. Similarly, if a machine learning model deployed for loan approval begins exhibiting bias against certain applicants, the template would facilitate thorough incident filing and feedback collection, maintaining transparency and accountability while guiding remediation efforts.

Key elements of Incident Playbook Template

Key elements of the Incident Playbook Template typically include: a clear incident description, date and time of occurrence, impacted systems or stakeholders, root cause analysis, immediate actions taken, escalation contacts and resolution status. The template should also provide sections for documenting lessons learned and capturing stakeholder feedback to inform future mitigation strategies and process improvements.

Trust report: Definition, structure and governance integration

A Trust report is a recurring, evidence-based summary that documents the operational status of an AI workflow, demonstrating that the system is functioning safely, as governed, and within established performance targets. Designed for both system-level and, optionally, organization-level (roll-up) perspectives, the Trust report typically spans 1–2 pages and serves as a primary artifact for internal and external stakeholders to assess the effectiveness of AI governance mechanisms.

Structure and purpose

At the system level, the Trust report consolidates information about a specific AI workflow, detailing compliance with controls (aligned with section 4.4) and real-world performance outcomes. The organization-level roll-up aggregates findings from multiple system-level reports, providing a holistic view of AI governance across the enterprise (as referenced in section 6).

Role of external assurance

To strengthen trust and transparency, the Trust report incorporates third-party validation as a form of external assurance. Independent assessors review report contents, verifying that controls are in place and functioning as intended, and that key performance indicators are met. This external review provides credible evidence for regulators and external auditors.

Key elements of the Trust report

- Ownership: Names the accountable owner(s) for the AI workflow and report generation.
- Controls: Documents control measures in place, referencing control layers (see Table 3).
- Performance vs. targets: Summarizes operational metrics compared to pre-defined targets.
- Incident counts: Reports the number and nature of incidents, including mitigations taken.
- Changes: Details significant changes to the system or controls since the last report.
- External assurance: Confirms the scope and outcomes of third-party validation activities.

EY | Building a better working world

EY is building a better working world by creating new value for clients, people, society and the planet, while building trust in capital markets.

Enabled by data, AI and advanced technology, EY teams help clients shape the future with confidence and develop answers for the most pressing issues of today and tomorrow.

EY teams work across a full spectrum of services in assurance, consulting, tax, strategy and transactions. Fueled by sector insights, a globally connected, multi-disciplinary network and diverse ecosystem partners, EY teams can provide services in more than 150 countries and territories.

All in to shape the future with confidence.

BMC Agency
GA 20125584

EYG No. 009538-25Gbl
ED None