Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.080888

REVIEW

From Documents to Decisions: Enterprise-Grade LLM Systems for Zero-Hallucination, Attributed Generation, and Regulatory Alignment

Yenjou Wang[1], **Chihtan Cheng**[2] **and Jia-Wei Chang**[3,*]

[1]Department of Information, Artificial Intelligence and Data Science, Daiichi Institute of Technology, Taito, Tokyo, Japan
[2]Ph.D. Program in Intelligent Engineering, National Taichung University of Science and Technology, Taichung City, Taiwan
[3]Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung City, Taiwan

*Corresponding Author: Jia-Wei Chang. Email: jiaweichang.gary@gmail.com

Received: 17 February 2026; Accepted: 20 April 2026; Published: 27 May 2026

ABSTRACT: As large language models (LLMs) become increasingly integrated into enterprise decision-making processes, structural pressures such as version drift, cross-source evidence integration, and regulatory accountability have shifted the primary challenge from isolated generative performance to system-level consistency, traceability, and governability. This paper systematically reviews key technological developments relevant to enterprise requirements, including document perception, retrieval-augmented generation (RAG), hybrid RAG-KG architectures, fine-grained attribution evaluation, and multi-agent coordination. The analysis demonstrates that the main obstacle to enterprise LLM adoption is not model capability, but rather the structural gap between fragmented technical modules and the need for high-reliability decision-making. In response, a risk-controlled data flywheel architecture is proposed that integrates perception, reasoning, verification, and governance layers. By converting reasoning outputs into observable risk signals and feeding them back into retrieval and structural components, this architecture establishes a continuous improvement loop. This approach provides a systematic deployment blueprint for enterprise-grade LLM systems, emphasizing traceability, accountability, and sustainable optimization in high-risk and long-term operational contexts.

KEYWORDS: Large language models (LLMs); retrieval-augmented generation (RAG); knowledge graph (KG); optical character recognition (OCR); enterprise AI systems; risk-controlled architecture; governance and compliance; attribution and faithfulness; multi-agent systems; data flywheel

1 Introduction

Large Language Models (LLMs) have undergone rapid evolution in recent years and are increasingly integrated into enterprise knowledge management and decision-support systems. As their capabilities expand, foundation models have generated significant discussion regarding systemic risks and governance implications [1,2]. In contrast to consumer-facing chatbot applications, outputs generated within enterprise environments can directly affect financial decisions, contract interpretation, regulatory analysis, and internal governance. Consequently, errors in generated content are not merely technical imperfections; they may lead to compliance violations, legal liabilities, and operational risks [3]. For example, in high-risk domains such as finance and healthcare, evolving regulatory requirements and compliance standards can significantly affect how information must be retrieved, interpreted, and validated, placing additional constraints on system

architecture and increasing the cost of erroneous outputs. The primary challenge in deploying LLMs in enterprise settings is therefore not merely to enhance generative performance, but to establish a risk-controlled system architecture that is trustworthy, auditable, and aligned with institutional requirements.

Enterprises often assume that connecting internal documents to a retrieval mechanism and implementing Retrieval-Augmented Generation (RAG) are sufficient to mitigate the risk of hallucination and ensure evidence-based responses. However, this assumption oversimplifies the complexity of enterprise data. Such data are highly heterogeneous, encompassing long-form PDFs, scanned contracts, financial statements, multi-version policy documents, and database records. These documents frequently contain cross-page dependencies, intertwined exception clauses, and loosely structured content that was not originally intended for generative models. In the absence of systematic document understanding and knowledge restructuring strategies, even models with retrieval capabilities may generate incorrect inferences due to fragmented evidence, truncated context, or indexing bias. In these scenarios, retrieval does not eliminate risk; rather, it may allow risk to propagate through poorly structured inputs and misaligned indexing processes.

Generative models are fundamentally probabilistic next-token prediction systems. Their training objective is to predict the most likely linguistic continuation, rather than to verify factual accuracy or maintain institutional consistency. Prior research demonstrates that LLMs may lack reliable self-calibration under uncertainty [4] and often rely on shortcut features during inference [5]. The tension between fluency and faithfulness in natural language generation has been systematically explored [6]. When presented with Out-Of-Distribution (OOD) data, incomplete evidence, or biased retrieval results, LLMs may generate fluent but factually incomplete or inaccurate responses. This phenomenon, known as hallucination, frequently manifests as partially correct content with critical omissions rather than entirely fabricated information. In enterprise contexts, such deviations can propagate through document processing, indexing, retrieval, and generation stages, ultimately resulting in institutional and accountability risks. At the system-level, hallucination should be understood not only as a model-level defect but as an indicator of inadequate cross-layer risk control.

Enterprise deployment of generative AI necessitates a transition from one-time model integration to continuous risk-control logic. As shown in Fig. 1, this governance approach aligns with the Plan-Do-Check-Act (PDCA) cycle. In the planning phase, organizations develop document-understanding and indexing strategies to ensure structured knowledge representation. During execution, RAG and constrained generation mechanisms support decision-making. In the checking phase, faithfulness evaluation, attribution verification, and hallucination detection determine whether generated outputs correspond with supporting evidence. In the action phase, validation results and audit signals are incorporated into knowledge structures and retrieval strategies to drive continuous improvement. In the absence of a closed-loop mechanism, generative systems are limited to static safeguards and cannot achieve sustained stability or self-improvement.

However, PDCA provides only a governance abstraction and does not fully capture the layered technical architecture and cross-layer risk-propagation mechanisms inherent in enterprise-grade LLM systems. Prior studies on foundation model risks and governance [1,2], along with analyses of broader societal and deployment impacts of generative AI systems [3], indicate that integrating LLMs into institutional workflows introduces multi-layered technical and organizational risks. Moreover, formal governance mechanisms alone are insufficient to close accountability gaps [7], and documentation tools, such as model cards, must be operationally embedded within system pipelines to support effective accountability [8]. Research in Machine Learning (ML) production systems further highlights the importance of readiness assessment and technical debt management for risk-controlled deployment [9], while corporate data science practices inherently

involve collaboration and distributed accountability structures [10]. Participation or procedural adjustments alone do not resolve structural risks in machine learning systems [11].

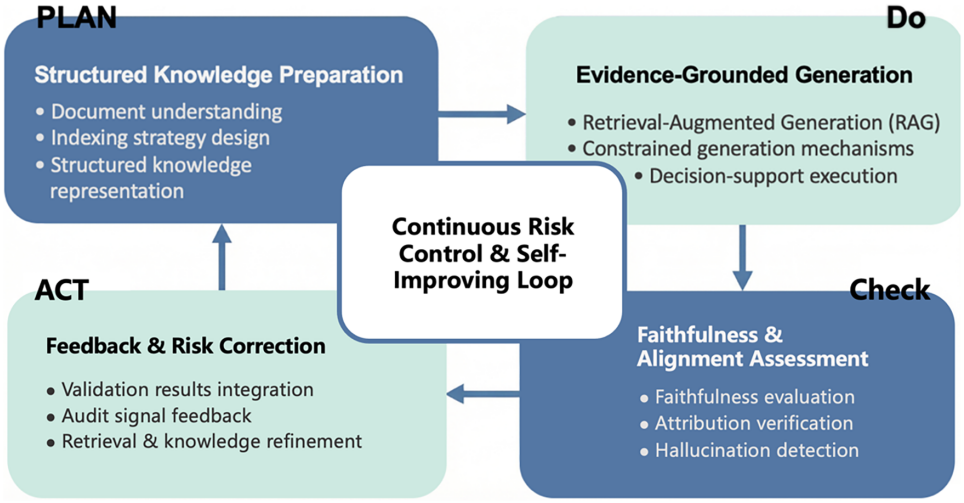

Figure 1: PDCA-based risk-control logic for enterprise LLM systems.

Consequently, the design of enterprise LLM systems should be treated as a systems-engineering challenge rather than a single-model optimization problem. Risk is not isolated within individual modules. Inconsistent data sources may distort indexing, indexing errors can diminish retrieval accuracy, and retrieval bias can heighten the probability of hallucination and citation drift. In the absence of robust verification and audit mechanisms, these challenges may escalate into institutional risks. Thus, risk propagates across document processing, indexing, retrieval, generation, and governance layers.

Fig. 2 presents the central conceptual framework of this paper: a risk-controlled, self-improving, enterprise-grade RAG architecture. Rather than treating RAG as an isolated technical module, the framework conceptualizes the system as a layered structure in which risks propagate forward and audit and compliance mechanisms provide reverse reinforcement signals. These feedback mechanisms enable upstream correction, structural stabilization, and continuous optimization. Therefore, Fig. 2 functions as a risk-oriented system blueprint for enterprise LLM deployment, rather than merely a technical pipeline.

This paper presents a systems-oriented review of enterprise LLM deployment. It synthesizes recent advances in document understanding, RAG, knowledge graph (KG) integration, and governance mechanisms. The review also examines practical challenges encountered during enterprise deployment, analyzes how these issues propagate across system layers, and discusses corresponding mitigation strategies. These elements are structured within a layered architecture to facilitate system-level analysis of enterprise workflows.

The main contributions of this paper are summarized as follows:

1. Enterprise LLM deployment is reframed as a cross-layer risk-propagation and risk-control problem rather than merely a model-performance issue.
2. A risk-controlled layered architecture that integrates document processing, hybrid retrieval, attribute generation, and governance auditing is presented to support risk control and continuous improvement.
3. Mechanisms for hallucination suppression, faithfulness evaluation, and regulatory alignment are systematically reviewed and structured into an operational risk-control framework for enterprise adoption of generative AI.

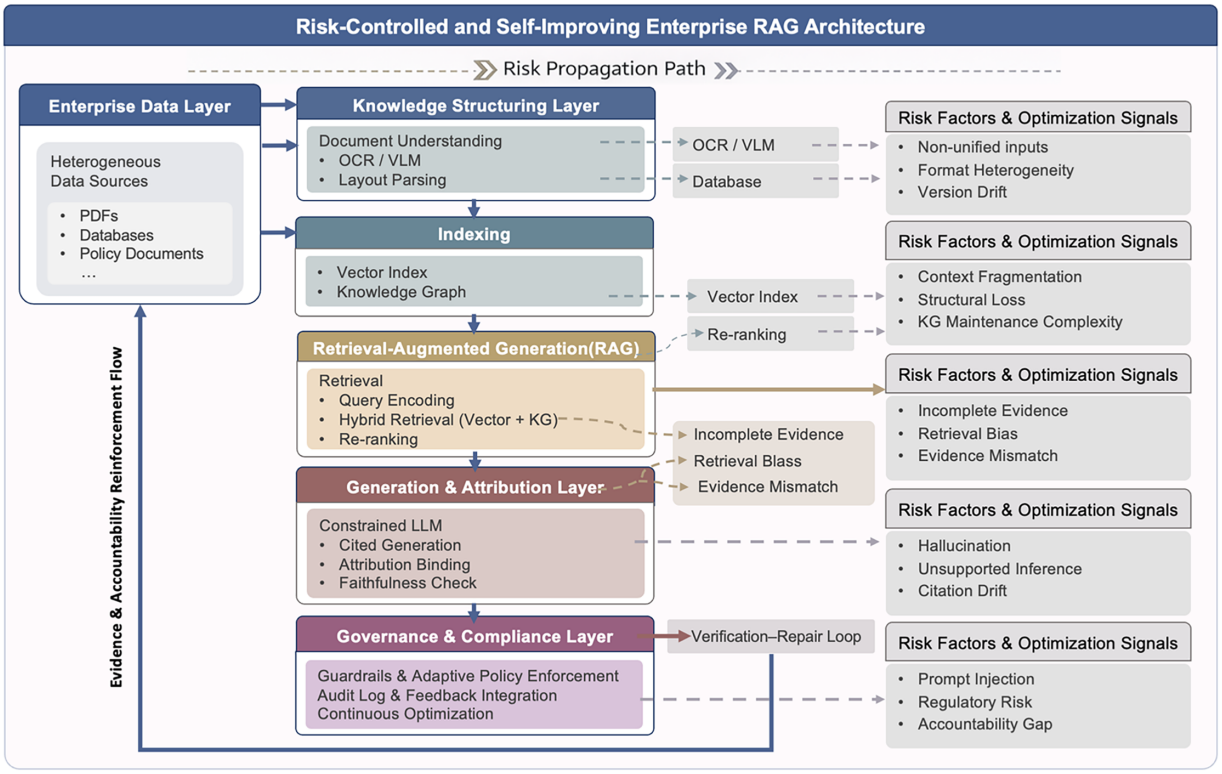

Figure 2: From documents to decisions: A layered architecture for enterprise-grade LLM systems.

The remainder of this paper is structured as follows. Section 3 examines foundational document-processing capabilities and their influence on downstream generation quality. Section 4 analyzes the design trade-offs of RAG architectures in enterprise contexts and discusses their progression toward agentic systems. Section 5 addresses evaluation methods for hallucination suppression and attributed generation. Section 6 explores security threats and regulatory issues in institutional environments. Section 7 integrates these components into an agentic workflow with guardrail mechanisms. Section 8 concludes by discussing the balance between risk control and self-improvement in enterprise LLM systems and outlines future research directions.

2 Background and Evolution of Enterprise LLM Systems

As LLMs are deployed in enterprise environments, challenges such as data heterogeneity, non-stationary knowledge, and regulatory constraints reveal limitations beyond model capability. These challenges prevent enterprises from relying solely on a single technical solution to address broader issues. Furthermore, recent discourse on institutional implementation and compliance evaluation underscores the necessity of post-deployment monitoring and accountability mechanisms [12].

The adoption of LLMs by enterprises constitutes a system-level process that extends beyond technical integration to include model design, data processing, retrieval strategies, evaluation mechanisms, risk-control procedures, and governance workflows. In real-world applications, effective AI systems must consider user behavior, contextual understanding, and mechanisms for continuous adaptation, rather than focusing solely on model performance [13]. Without ongoing optimization and feedback, maintaining long-term stability and institutional trust is challenging. Consequently, enterprise-grade LLM development has shifted from a model-centric approach to an integrated design that incorporates knowledge enhancement,

reliability evaluation, and governance practices. This shift establishes the conceptual basis for the layered analysis and workflow presented in the following sections.

2.1 From Model-Centric Intelligence to Knowledge-Augmented Systems

With the continuous expansion of pre-training scale and parameter size, models achieved significant improvements in language understanding, generation, and cross-task transfer performance [14,15], demonstrating strong linguistic fluency and reasoning ability. This progress led to impressive results in open-domain question answering and general dialogue tasks, reinforcing a model-centric perspective in which increasing model capacity was regarded as sufficient to solve practical problems.

However, when deployed in enterprise decision-making and knowledge management contexts, the limitations of such models gradually become evident. Enterprise knowledge is not a static corpus but is distributed across internal documents, proprietary databases, and continuously updated regulations. Parametric memory cannot capture real-time data changes or clearly indicate the source of generated content. In compliance-sensitive environments, linguistic plausibility does not necessarily imply verifiable correctness. If outputs cannot be traced to their sources, even linguistically strong models are difficult to incorporate into formal decision-making processes.

Against this background, knowledge augmentation mechanisms have gradually become a necessary design choice. By introducing external document retrieval into the generation process, RAG seeks to compensate for the limitations of parametric knowledge [16], enabling generated content to be linked to explicit data sources. Subsequent studies have further investigated the integration of retrieval and generation and its influence on reasoning processes [17] and have analyzed how retrieval quality and ranking strategies affect final outputs in black-box language model settings [18]. These works indicate that RAG is not merely a tool for knowledge supplementation but a system-level design choice that reshapes the generative process.

Nevertheless, the introduction of retrieval mechanisms does not guarantee reliability. Ranking bias in retrieved results, improper document chunking strategies, and contextual discontinuities across documents may introduce new sources of error during generation. In other words, while knowledge augmentation improves knowledge coverage, it also increases system-level complexity, extending challenges from the model layer to data processing and retrieval layers. This evolution further lays the foundation for subsequent developments in reliability evaluation and governance mechanisms.

2.2 Reliability, Evaluation, and Accountability

Although knowledge-augmented architectures incorporate external evidence sources into generative models, the presence of retrieval does not necessarily ensure verifiability. In practical deployments, even when systems can provide cited documents, generated content may still exhibit evidence misalignment, cross-paragraph reasoning errors, or incomplete correspondence between citations and conclusions. This phenomenon reveals a deeper issue: retrieval improves the accessibility of information sources, but it does not fundamentally alter the probabilistic and uncertain nature of the generation process.

Studies have shown that language models may produce fluent responses at the linguistic level while generating factually incorrect or misleading content [19]. Broader analyses further indicate that hallucination has become a central risk for generative models across multiple task settings [20]. Under cross-paragraph and multi-document conditions, models may incorrectly integrate information due to limitations in compositional understanding [21]. Even when retrieval mechanisms are introduced, models may fail to consistently follow retrieved evidence when conflicts arise between parametric knowledge and external sources [22].

In addition, generative models often exhibit unstable calibration under uncertainty, and their confidence scores do not necessarily reflect actual correctness rates [23]. This implies that, in enterprise environments, relying solely on model outputs or confidence scores is an unreliable basis for decision-making. When such deviations are embedded in enterprise decision-making processes, the issue is no longer whether a single answer is correct but how errors propagate across document processing, retrieval, and generation stages, ultimately creating institutional and accountability risks.

Therefore, the development of enterprise-grade LLM systems has gradually shifted from the question of how to expand knowledge coverage to how to ensure that generated content is verifiable and traceable. Reliability is no longer regarded as a byproduct of model performance but as a core property that must be actively constructed through system design and evaluation procedures. This transition further lays the theoretical and practical foundation for the subsequent discussion of risk-control and governance architectures.

2.3 Toward Integrated Enterprise Workflows

As knowledge-augmentation and evaluation mechanisms are gradually incorporated into the generative pipeline, the challenge for enterprise-grade LLM systems is no longer limited to whether the model itself is accurate but also whether the overall architecture remains stable. Generative models are embedded within multi-stage workflows that include document parsing, content chunking, vector index construction, retrieval and re-ranking, generation control, and post-hoc verification. Individual modules may perform well under isolated testing conditions, yet once coupled together, their errors may propagate and amplify within the system.

Data-layer issues often initiate risk propagation. Insufficient data validation and quality control can undermine downstream models [24]. In high-risk settings, data dependencies and workflow discontinuities may produce so-called "data cascades," allowing errors to gradually spread across organizational and technical layers [25]. These phenomena indicate that data consistency and structural quality influence not only model outputs but also overall system stability.

Moreover, if testing and monitoring mechanisms focus solely on model-level metrics while neglecting interactions between data flow and retrieval layers, cross-layer risks may go undetected. Research on machine learning testing emphasizes that the testing scope should encompass data, model, and process levels [26]. In practical deployment, industry guidance further highlights the importance of continuously monitoring model behavior, data drift, and prediction quality to maintain long-term system stability [27]. Without systematic deployment strategies and organizational coordination mechanisms, enterprises often encounter hidden technical debt and operational pressures in later stages of adoption [28].

For large language models, deployment complexity is even higher. Recent studies on LLM serving indicate that operational integration presents challenges related to memory consumption, inference cost, latency control, throughput optimization, and coordination among model, data, and system-level components [29]. These observations suggest that LLM adoption is not merely a model replacement but an architectural engineering task that requires coordination across data flow, retrieval, and generation layers. Accordingly, enterprise-grade LLM design must shift from single-model optimization toward system-level architectural thinking. Risk is therefore understood not as a localized error but as a cross-layer propagation phenomenon that requires coordinated mechanisms for validation, monitoring, and integration. This transition from model-centric to system-oriented design provides the foundation for the layered architecture and risk-control framework discussed in the subsequent sections.

3 Foundational Document-Oriented Capabilities: From PDFs to Searchable Knowledge

The rapid advancement of LLMs has established document-oriented understanding as a foundational capability in enterprise AI systems. Critical enterprise knowledge is often stored in PDFs, scanned documents, and structurally complex file formats that cannot be directly integrated into retrieval and reasoning pipelines. Consequently, converting static documents into searchable and traceable knowledge representations is a prerequisite for enterprise LLM deployment and serves as a primary risk-control measure at the data layer [30].

Traditional document processing pipelines generally begin with Optical Character Recognition (OCR), followed by layout analysis and rule-based post-processing. As document formats become more complex, including multi-column layouts, dense tables, and mixed text-image structures, OCR-centered workflows exhibit structural limitations. OCR focuses on text extraction rather than semantic interpretation and therefore cannot reliably capture relational meaning across charts, tables, or distributed paragraphs. Partial understanding of document structures can introduce early-stage errors that affect indexing and retrieval quality [31].

Therefore, improving OCR accuracy alone does not address the challenges of cross-scenario generalization and higher-level semantic interpretation. Recent research has integrated vision-language models into document understanding pipelines to complement OCR outputs. By combining visual context with language reasoning, these models advance document processing from basic text recognition to content-level understanding [30].

Based on this background, this section examines the core capabilities required to transform PDFs into searchable knowledge. Section 3.1 discusses representative OCR approaches and engineering trade-offs. Section 3.2 reviews vision-language models and layout analysis techniques and clarifies their roles in next-generation document understanding systems.

3.1 OCR-Based Document Understanding: Capabilities, Limits, and Engineering Trade-Offs

OCR has long served as the perception layer in document understanding systems. Its primary role is to convert image-based documents into machine-readable text and initial structural information for downstream processing. In enterprise AI systems, OCR serves as an entry module that connects physical documents with digital knowledge-processing pipelines [32].

As enterprise documents increasingly exhibit cross-page dependencies, multi-column layouts, table-dense content, and mixed text-image structures, the capabilities and engineering trade-offs of OCR systems have become central determinants of overall document understanding quality and system stability. From a practical enterprise perspective, OCR must therefore be examined along two dimensions: the scope of its achievable capabilities and the constraints imposed at the engineering level. This analysis clarifies why OCR has gradually shifted from being treated as a core processing component to being positioned as a lower-level perceptual module that requires supplementation and constraints within the broader architecture.

3.1.1 Capabilities of OCR Systems in Structured Document Processing

Document-oriented understanding has long relied on OCR as the primary mechanism for digitizing text. Its core function is to convert visually represented documents into machine-readable text for integration into retrieval and knowledge-processing workflows. However, OCR focuses on text extraction rather than semantic interpretation, and its support for higher-level structural relationships remains limited. Early research formalized OCR as a modular pipeline comprising preprocessing, segmentation, classification, and post-processing, establishing a deployable engineering framework that underpins enterprise document

digitization [32]. In practice, mainstream OCR systems follow this modular design and enhance performance through engineering optimization. Representative engines such as Tesseract integrate adaptive thresholding and layout analysis to determine text regions and reading order, reflecting the traditional segmentation-and-classification paradigm [33]. Under structured conditions, such as stable layouts and printed text, OCR provides efficient and consistent recognition, supporting enterprise digitization and full-text retrieval.

Functionally, OCR converts textual elements into structured units that can be electronically indexed and processed [34]. Nevertheless, recognition robustness declines as document quality deteriorates or layouts become complex, particularly in low-resolution or unstructured scenarios [35]. Consequently, recognition errors and reading-order inconsistencies may degrade downstream indexing quality.

To mitigate these issues, later research incorporated reading-order modeling and layout-aware mechanisms. Methods such as TextScanner model sequential relationships among textual elements to improve output coherence [36], while layout-aware approaches help correct spatial misalignment in multi-column documents [37]. In addition, contextual post-correction using transfer learning and NLP techniques has been introduced to address character omissions and spelling errors [38]. Although these approaches improve performance in specific domains, their effectiveness still depends on data distribution alignment.

OCR provides mature, engineerable capabilities for structured document processing, particularly in stable-layout enterprise scenarios. However, its core objective remains text extraction rather than semantic interpretation. As document structures become more diverse, OCR may recognize textual elements in charts and tables but cannot infer higher-level meaning or relational context. When these perceptual limitations are embedded in enterprise workflows, early extraction errors can affect subsequent retrieval and generation stages, making OCR insufficient as a standalone foundation for systems that require traceability and semantic consistency.

3.1.2 Engineering Limits and Error Propagation in OCR-Based Pipelines

Recent advances in OCR have been primarily driven by deep learning models. For instance, TrOCR employs an end-to-end Transformer-based architecture that improves generalization to handwritten text and non-standard fonts, demonstrating progress beyond the limitations of traditional character classifiers [38]. Despite these improvements, core challenges persist in enterprise deployment. Document sources in real-world applications are highly heterogeneous, with ongoing variations in scanning quality, layout formats, and data distributions. Consequently, enterprise systems continue to depend on transfer learning and domain adaptation strategies to address such variability [38]. OCR systems, therefore, remain reliant on context-specific engineering configurations rather than achieving robust cross-domain adaptability. In the absence of continuous monitoring or adjustment mechanisms, shifts in data conditions can transform recognition errors from isolated technical issues into significant risks for downstream knowledge processing workflows.

OCR architectures function as multi-stage perceptual pipelines, in which errors introduced in early stages can be amplified by subsequent modules, resulting in nonlinear error propagation. Prior research demonstrates that character- and word-level OCR errors impact information retrieval and text mining performance, leading to biased results and semantic misinterpretation, while also distorting document representation during structural transformation and indexing and degrading downstream retrieval and reasoning [39].

Fig. 3 provides a practical example of error propagation within an OCR-RAG integrated workflow. If a key entity, such as the address number “No. 10,” is misrecognized as “No. IO,” this minor character-level error can result in entity linking failures or incorrect vector representations during indexing. Since RAG systems depend on structured indexing and semantic-similarity retrieval, such errors may prevent

the retrieval of correct documents. Even with an accurate user query, the system may produce incorrect conclusions. This example highlights the structural risks present in OCR-based pipelines for enterprise document understanding and demonstrates that improving recognition accuracy alone is insufficient to ensure reliability in downstream RAG and reasoning modules.

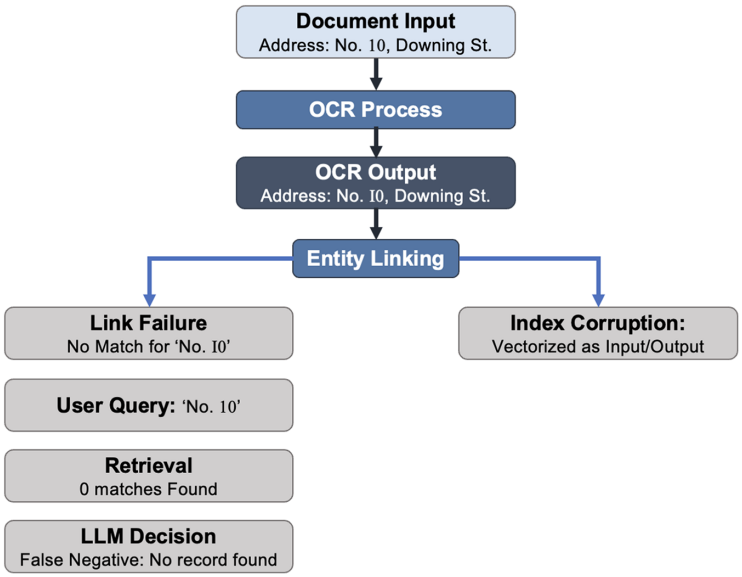

Figure 3: Workflow of OCR error propagation in RAG systems.

The challenges are exacerbated when processing long tables and complex layouts. If OCR fails to maintain table boundaries, row-column relationships, or hierarchical structures, the transformation into structured data can result in semantic misalignment, such as mismatches between numerical values and column labels. These errors may cause irreversible distortions in analytical outcomes [40]. Although such issues may not be immediately apparent in character-level accuracy metrics, they can significantly impact downstream decision-support systems.

The accumulation of errors becomes more significant when OCR outputs are integrated with large language models. Systematic analyses reveal a non-linear negative correlation between OCR noise levels and RAG system performance [41]. When knowledge bases contain excessive OCR-induced distortions or structural inconsistencies, even accurately retrieved text may provide unreliable evidence for language model reasoning, increasing the risk of hallucinated outputs. These risks undermine the traceability often attributed to RAG systems and challenge the accountability and reliability standards required for enterprise-grade deployments.

In summary, while OCR is essential for document digitization and initial structural conversion, its error propagation, maintenance demands, and limited cross-domain generalization constrain its ability to support reasoning- and decision-oriented enterprise knowledge systems. These limitations have motivated research toward architectures that jointly model visual layout and semantic structure, including Vision-Language Models (VLMs) and OCR-free document understanding methods.

3.2 The Development of VLM and Layout-Aware Document Parsing

Even when high-performance OCR systems produce stable text, the lack of holistic modeling of document structure and layout semantics can still lead downstream components, such as RAG or enterprise decision-support systems, to produce semantic deviations or erroneous inferences due to structural misinterpretation.

In this context, layout-aware modeling, VLMs, and OCR-free end-to-end document understanding architectures have emerged as important directions for advancing document-oriented understanding beyond text-level correctness toward more reliable content-level interpretation.

3.2.1 Layout-Aware VLM: Addressing Structural Blind Spots of OCR

Despite improvements in character-level accuracy, traditional OCR architectures are still limited in their ability to model spatial structure and semantic roles within documents. Accurate text recognition does not guarantee correct interpretation of layout relationships or document hierarchy. In complex layouts, such as multi-column or table-heavy documents, relying solely on textual information can lead to structural misinterpretation and semantic drift in subsequent processing steps.

This limitation has prompted a shift in document understanding research toward multimodal representations that integrate both layout and visual context. To address these challenges, models such as LayoutLM incorporate two-dimensional spatial information during language model pretraining, which enables joint learning of textual content and layout relationships [30].

LayoutLM builds on BERT by representing the spatial distribution of text blocks with two-dimensional positional embeddings and integrating visual features to capture font style and appearance. During pretraining, the model uses masked visual-language modeling and multi-label document classification objectives, enabling self-supervised learning on large-scale scanned document corpora. Experimental results show that LayoutLM significantly outperforms text-only or image-only models on tasks such as form understanding, receipt information extraction, and document image classification, underscoring the importance of layout awareness in document-level semantic modeling.

Building on this foundation, LayoutLMv2 incorporates visual features directly into the pretraining stage rather than introducing them only during fine-tuning and adopts a dual-stream multimodal Transformer architecture to model cross-modal interactions among text, layout, and image representations, thereby improving representational alignment and consistency across-modalities [42]. Subsequently, LayoutLMv3 simplifies the architecture by replacing conventional CNN-based visual backbones with a unified Transformer framework that jointly processes text tokens and image patches. Through a unified masked pretraining objective, LayoutLMv3 achieves state-of-the-art performance across multiple document understanding benchmarks, demonstrating the potential of end-to-end multimodal modeling to balance performance and engineering simplicity [43].

Alongside the LayoutLM series, DocFormer represents another important research direction. It uses a single Transformer architecture to jointly model textual content, visual features, and spatial information, and employs multimodal self-attention mechanisms to enable collaborative modeling across-modalities. Experimental results show that, despite a relatively small parameter count, DocFormer achieves competitive or superior performance across multiple document understanding tasks, highlighting the practical value of layout awareness and multimodal fusion in real-world applications [44].

Layout-aware VLMs do not replace OCR entirely but instead serve as upper-layer enhancement modules that compensate for its limitations in structural understanding and semantic modeling. Documents are thus treated not merely as collections of text but as structured entities with relational and semantic coherence.

3.2.2 OCR-Free Architectures: End-to-End Document Understanding

With the maturation of multimodal models, research attention has gradually shifted from improving OCR outputs to questioning the need to maintain the traditional OCR pipeline. The OCR-2.0 framework argues that conventional OCR systems rely on multi-stage pipelines involving preprocessing, segmentation, recognition, and post-processing [45]. Such modular designs are prone to error accumulation, increased engineering complexity, and greater maintenance costs. In contrast, OCR-2.0 advocates a unified end-to-end model that jointly performs perception and structured output generation, treating document understanding as an integrated learning problem rather than a sequential engineering workflow.

This perspective is further exemplified by Donut [46]. Donut adopts a pure Transformer architecture that directly maps document images to structured outputs, such as JSON or key-value pairs, thereby performing both text reading and document understanding within a single model and bypassing the conventional OCR process. Experimental results show that Donut reduces the error propagation commonly observed in OCR-based architectures and demonstrates robust generalization across multilingual and complex layout documents. These findings suggest that OCR-free approaches are technically feasible for document understanding tasks.

However, OCR-free methods present additional considerations such as explainability, data dependency, and deployment cost, which necessitate thorough evaluation in enterprise contexts. These challenges have led to the creation of evaluation benchmarks and engineering metrics for the systematic assessment of document understanding architectures.

3.2.3 Evaluation Perspectives and Engineering Realities in Document Understanding

With the transition from modular OCR pipelines to end-to-end multimodal architectures, evaluation metrics have expanded beyond character-level accuracy. In enterprise applications, significant risks include structural misinterpretation, field misalignment, and failures in cross-sectional semantic integration. These issues are not addressed by character-level metrics, yet they have a direct impact on tasks such as contract interpretation, financial analysis, and auditing. As a result, recent research has increasingly focused on document-level structural and semantic correctness.

The DocVQA benchmark [47] reflects this transition by requiring models to integrate visual layout and textual content when answering document-level questions. This approach promotes a shift from component-level extraction to holistic document comprehension. Subsequent studies [48] indicate that models that leverage layout features to infer field semantics are particularly effective for highly structured documents, such as audit reports, contracts, and financial statements. This finding highlights the practical significance of layout-aware modeling in enterprise and compliance contexts. More recently, evaluation frameworks such as olmOCR-Bench [49] have introduced metrics focused on overall document structural correctness, facilitating systematic comparisons across models under diverse layout and domain conditions.

Table 1 demonstrates that models such as Infinity-Parser-7B and anchored prompting variants of olmOCR exhibit clear advantages in structurally complex scenarios, including multi-column layouts and legacy scanned documents. These findings suggest that integrating layout awareness with contextual constraints substantially enhances document parsing quality. In contrast, traditional OCR-based pipelines continue to exhibit limitations in structural consistency, indicating that character recognition and rule-based post-processing alone are inadequate for comprehensive document-level understanding.

Table 1: Performance comparison of different models on the olmOCR benchmark across diverse document domains and structural challenges.

Model	Overall	ArXiv	Old Scans Math	Tables	Old Scans	Headers & Footers	Multi Col.	Long-Tiny Text	Base
GOT OCR	48.3	52.7	52.0	0.2	22.1	93.6	42.0	29.9	94.0
Marker v1.6.2	59.4	24.3	22.1	69.8	24.3	87.1	71.0	76.9	99.5
MinerU v1.3.10	61.5	75.4	47.4	60.9	17.3	96.6	59.0	39.1	96.6
Mistral OCR API	72.0	77.2	67.5	60.6	29.3	93.6	71.3	77.1	99.4
GPT-4o (No Anchor)	68.9	51.5	75.5	69.1	40.9	94.2	68.9	54.1	96.7
GPT-4o (Anchored)	69.9	53.5	74.5	70.0	40.7	93.8	69.3	60.6	96.8
Gemini Flash 2 (No Anchor)	57.8	32.1	56.3	61.4	27.8	48.0	58.7	84.4	94.0
Gemini Flash 2 (Anchored)	63.8	54.5	56.1	72.1	34.2	64.7	61.5	71.5	95.6
Qwen 2 VL (No Anchor)	31.5	19.7	31.7	24.2	17.1	88.9	8.3	6.8	55.5
Qwen 2.5 VL (No Anchor)	65.5	63.1	65.7	67.3	38.6	73.6	68.3	49.1	98.3
olmOCR v0.1.68 (No Anchor)	76.3	72.1	74.7	71.5	43.7	91.6	78.5	80.5	98.1
olmOCR v0.1.68 (Anchored)	77.4	75.6	75.1	70.2	44.5	93.4	79.4	81.7	99.0
Infinity-Parser-7B	82.5	84.4	83.8	85.0	47.9	88.7	82.4	86.4	99.8

Note: Adapted from Wang et al. (2025) [49].

Table 2 further summarizes the evolving differentiation between OCR and VLM-based approaches within enterprise systems. OCR continues to serve as an effective foundational layer for document digitization and initial text extraction, especially in structured and stable layouts. In contrast, VLM-based models increasingly operate as an intermediate semantic layer, bridging document parsing, structural modeling, and knowledge construction. This development reflects a broader transition in document understanding, moving from the extraction of readable text to the generation of representations that are searchable, inferable, and suitable for decision support.

Table 2: Role differentiation and design trade-offs between OCR and VLM in enterprise document understanding systems.

Comparison Dimension	Traditional OCR	Vision Language Model (VLM)
Core Design Objective	Converts document images into editable and searchable textual representations.	Jointly models visual layout and linguistic semantics to interpret document content and structure.
System Role	Serves as a low-level perception module.	Functions as a high-level document understanding and semantic modeling component.
Processing Unit	Operates at the level of characters, tokens, and text lines.	Processes document blocks, layout structures, and cross-modal semantics.
Handling of Layout Structure	Relies on explicit segmentation and rule-based layout analysis, which is sensitive to layout variations.	Implicitly models layout and spatial relationships through learning-based mechanisms.
Support for Tables and Complex Structures	May lose row and column relationships and typically requires additional table parsing modules.	Treats tables as integrated visual and linguistic structures for joint understanding and reasoning.
Semantic Understanding Capability	Provides textual output only, while semantic understanding depends on downstream components.	Possesses language reasoning ability and can directly generate semantic-level representations.

(Continued)

Table 2 (continued)

Comparison Dimension	Traditional OCR	Vision Language Model (VLM)
Error Characteristics	Multistage pipeline in which errors may propagate and amplify across stages.	End-to-end modeling that can absorb perceptual noise to a certain extent.
Cross Scenario Generalization	Requires transfer learning or rule adjustment for adaptation.	Demonstrates stronger zero-shot or few-shot capability for new document types.
Relationship with RAG	Provides textual foundations for retrieval but may introduce structural and semantic noise.	Produces semantically consistent and structure aware knowledge units, thereby improving evidence quality in RAG systems.
Ideal Position in Enterprise Systems	Serves as a preprocessing layer for document digitization and initial text extraction.	Acts as a key intermediary layer for document interpretation, semantic enrichment, and knowledge construction.

Recent advances, including layout-aware VLMs and OCR-free architectures, aim to improve the conversion of static documents into machine-interpretable representations. However, when these advances are integrated into enterprise QA and decision workflows, document understanding alone does not ensure traceability or verifiability. Furthermore, with generative models involved, the focus shifts from parsing accuracy to ensuring evidence-supported outputs. As a result, this shift is driving the adoption of RAG as a core component in enterprise LLM systems.

4 RAG and Agentic Pipelines

While LLMs are powerful, their factual accuracy and knowledge recency are constrained by fixed training data. In high-risk enterprise scenarios (e.g., contracts, financial reports), purely generative models lack source attribution and auditable reasoning, and hallucinations may lead to tangible risks, such as misleading financial decisions [50,51].

RAG addresses these limitations as a structural intervention by incorporating external evidence into the generation process, thereby improving accuracy, traceability, and controllability. Grounding outputs in explicit evidence reduces unsupported inference and functions as a knowledge mediation layer for auditable decision-making. Its effectiveness, however, depends on document understanding (OCR/VLM) and knowledge structuring (e.g., KG), which convert unstructured data into retrievable and verifiable assets. The following sections examine the evolution of RAG from basic retrieval pipelines to structured and agentic systems under enterprise constraints.

4.1 Hybrid Retrieval and Re-Ranking Techniques

In RAG, retrieval quality critically determines output faithfulness and attribution. However, in enterprise settings with growing knowledge bases and diverse queries, single retrieval methods often fail to balance recall and precision. Furthermore, dense retrievers underperform in zero-shot or OOD settings, which limits reliability in heterogeneous systems [52,53]. To address these challenges, this section examines hybrid retrieval and deep re-ranking to improve integration and coverage. In enterprise deployments, these approaches also serve as risk-mitigating mechanisms that constrain downstream generation variability.

4.1.1 The Core of Retrieval: Complementary Strengths of Sparse and Dense Models

In the RAG architecture, the first-stage task is to efficiently retrieve a set of semantically relevant candidate passages from a large corpus. Two primary retrieval strategies are commonly employed at this stage: sparse retrieval and dense retrieval, each with its own advantages.

A representative method for sparse retrieval is BM25, which relies on term frequency-inverse document frequency (TF-IDF) and other lexical statistics to perform exact matching. This approach is efficient for queries containing domain-specific terms or precise keywords [54]. More recently, SPLADE has emerged as a learning-based sparse retrieval method that constructs sparse term vectors with learned weights, thus combining the efficiency of sparse retrieval with improved semantic generalization [55].

On the other hand, dense retrieval methods such as Dense Passage Retrieval (DPR) [56] and E5 [57] employ deep neural networks to encode queries and documents as dense vectors and perform semantic matching in vector space via Approximate Nearest-Neighbor (ANN) search. These models are particularly suitable for fuzzy or conceptually synonymous queries. However, while bi-encoder models like E5 are known for their scalability and performance, they may overlook concrete entity names due to a lack of fine-grained token-level alignment [58]. This tension between semantic abstraction and lexical grounding has prompted further exploration of hybrid approaches that combine sparse and dense retrieval methods.

4.1.2 Hybrid Retrieval and Ranking Fusion

To build a high-recall, fault-tolerant Candidate Evidence Set before using costly generative models (LLMs) or deep re-ranking modules (such as cross-encoders), modern retrieval systems must overcome the limitations of relying on a single retrieval signal.

(a) Complementarity and Robustness

No single retrieval mechanism works for all cases. Probabilistic models such as BM25 have strong stability with OOD queries but lack deep semantic understanding. Neural retrieval models excel at semantic matching, but they may fail in zero-shot settings or under adversarial perturbations.

Hybrid retrieval is more than a way to improve accuracy. It acts as a diversity-based defense. BM25's rigid lexical matching offsets the hallucination risks of neural models.

(b) Addressing Score Heterogeneity

Because BM25 produces unbounded relevance scores derived from probabilistic assumptions, while dense retrievers output normalized similarity values (e.g., [0, 1]), direct score-level aggregation is often unstable and difficult to calibrate [54]. To address this mismatch, Cormack et al. proposed Reciprocal Rank Fusion (RRF), which performs fusion solely at the ranking level:

$$RRFscore\,(d) = \sum \frac{1}{k + r\,(d)} \tag{1}$$

By ignoring absolute score magnitudes, RRF rewards documents that are consistently ranked highly across multiple retrieval models and suppresses isolated noise, enabling the construction of a high-recall and noise-resistant candidate set prior to expensive downstream inference [59].

By fusing lexical and semantic signals with RRF, the system builds a high-recall, noise-resilient candidate set before inference. Such rank-level fusion also enhances traceability, as consensus-based selection reduces reliance on opaque similarity scores from a single model. This acts as a multi-expert consultation layer. It ensures downstream LLM reasoning is grounded in evidence that is both trustworthy and comprehensive.

4.1.3 Deep Re-Ranking Mechanisms

After hybrid retrieval ensures sufficient recall in the candidate set, deep re-ranking models serve as a final, precision-oriented layer. They perform fine-grained relevance judgment. Unlike first-stage retrieval, which relies on vector similarity, deep re-rankers model queries and documents together to capture token-level interactions and compositional semantics.

Discriminative cross-encoders based on BERT set the upper bound for neural re-ranking performance. They concatenate queries and documents in a single input sequence and perform binary relevance classification. These models achieve much higher accuracy than bi-encoder retrievers on large-scale benchmarks such as MS MARCO [60]. To improve robustness in low-resource or domain-shifted settings, MonoT5 reformulates ranking as a sequence-to-sequence generation task. It uses the logits of generated relevance tokens as scores and demonstrates strong data efficiency and zero-shot transfer capability [61].

Despite their effectiveness, deep re-rankers incur high computational costs that scale linearly with the number of candidates. To balance efficiency and accuracy, late-interaction models like ColBERT encode queries and documents independently. They preserve token-level representations, enable offline indexing, and reduce query latency with competitive effectiveness [58].

Re-ranking is not a replacement for Retrieval within the RAG stage, but a cost-aware, high-precision relevance adjudicator applied after recall-oriented pruning. In advanced enterprise systems, relevance judgments from re-ranking stages can further serve as feedback signals for iterative retrieval refinement.

4.1.4 Vector Database Selection and Enterprise Considerations

From an enterprise perspective, hybrid retrieval and deep re-ranking are coordinated system-level decisions constrained by latency, scalability, and cost. In practice, sparse and dense retrievers are combined to maximize recall, while expensive re-rankers are applied after candidate pruning, reflecting a quality-efficiency trade-off in production RAG systems.

The indexing module operationalizes this consolidation by enabling fast top-K access, hybrid query execution, and continuous index updates over evolving knowledge bases. Accordingly, vector database selection is driven less by peak retrieval accuracy than by the ability to support hybrid retrieval pipelines, cost-aware re-ranking, and enterprise requirements such as maintainability and observability [62]. Fig. 4 summarizes how hybrid retrieval, ranking fusion, and deep re-ranking are integrated into a deployable pipeline under these constraints. Beyond efficiency considerations, this integration increasingly reflects a governance-oriented design, in which the retrieval architecture is aligned with compliance, observability, and controllable generation behavior.

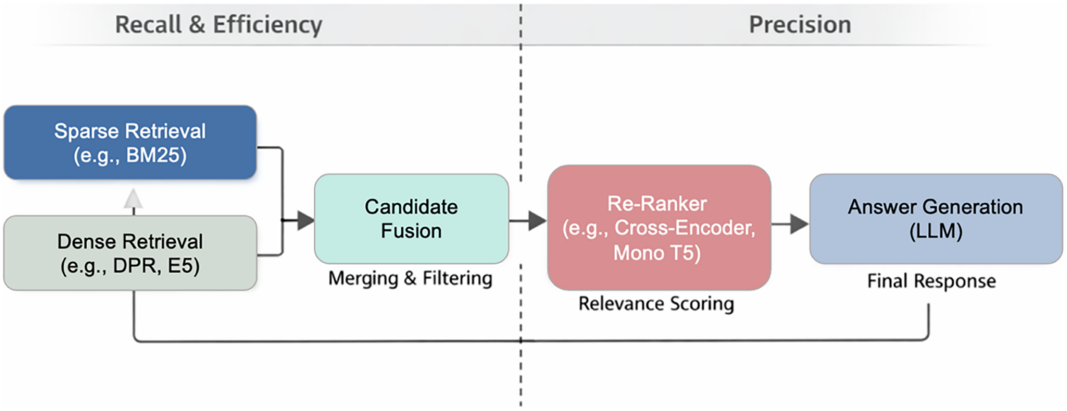

Figure 4: Hybrid retrieval and deep re-ranking pipeline in enterprise RAG systems.

4.2 Pipeline Variants and Optimization Strategies

While early RAG systems focused on simple vector-based retrieval, enterprise needs for high-precision and complex reasoning have led to a structural evolution in RAG architectures. As shown in Table 3, the move from Basic RAG to specialized types like GraphRAG, FileRAG, and Multi-hop RAG is not just about different methods but a logical response to specific structural flaws in standard pipelines, such as losing relational context, document fragmentation, and the inability to do iterative reasoning.

Table 3: Comparison of representative RAG pipeline variants.

Variant	Retrieval Unit	Knowledge Organization	Reasoning Support	Typical Use Cases
Basic RAG (Section 4.2.1)	Text chunk	Flat	Single-hop factual reasoning	FAQ, factual QA
GraphRAG (Section 4.2.2)	Node/edge	Graph-structured	Relational and multi-step reasoning	Regulations, policies
FileRAG (Section 4.2.3)	Document/ section	Hierarchical	Long-context aggregation and reasoning	Reports, manuals
Multi-hop RAG (Section 4.2.4)	Stepwise evidence	Dynamic	Explicit multi-hop reasoning	Analysis, synthesis

Fig. 5 presents an overview of common RAG pipeline components and representative variants. Regardless of the variant, all RAG systems rely on a large language model for the final generation step. The key differences lie in how evidence is retrieved, structured, and provided to the model. Early RAG architectures typically use a flat retrieval setting, in which a fixed top-K set of text chunks is retrieved and directly passed to the generator [62]. This design works well for simple factual queries but becomes less effective when relational reasoning, long document context, or multi-step evidence composition is required.

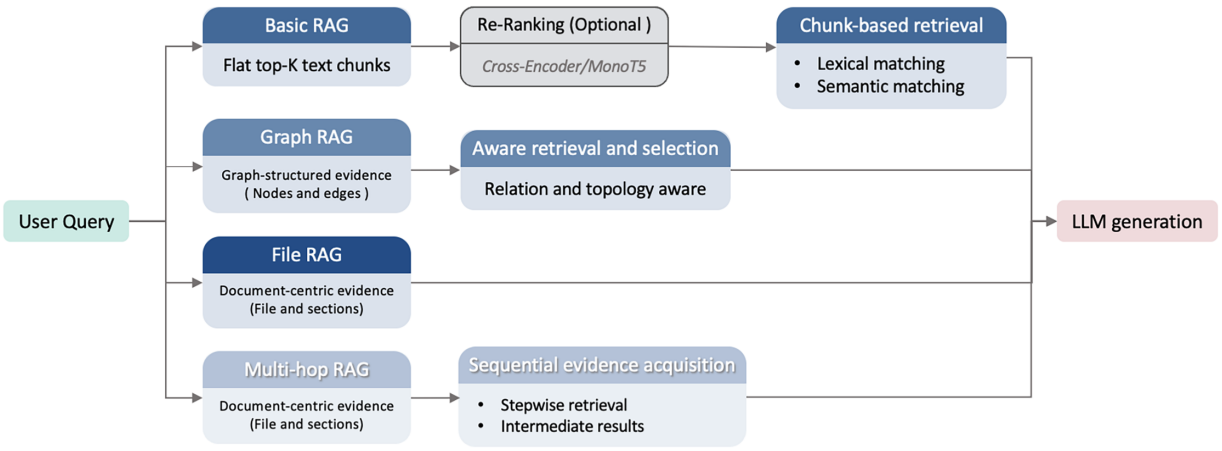

Figure 5: Architectural variants of RAG and their evidence retrieval structures.

To overcome these limitations, several structured RAG pipeline variants have been proposed. Graph RAG introduces graph-structured knowledge representations that support relation-aware evidence selection and reasoning over interconnected entities or clauses [63,64]. File RAG focuses on document-centric

retrieval for long, heterogeneous files, aiming to preserve contextual integrity by retrieving documents or sections rather than isolated text chunks [65,66]. Multi-hop RAG further extends this idea by decomposing complex queries into a sequence of retrieval steps, allowing intermediate results to guide subsequent evidence acquisition and synthesis [67].

Table 3 summarizes representative RAG pipeline variants by their retrieval units, knowledge organization strategies, supported reasoning patterns, and typical application scenarios. These variants differ primarily in the form of evidence they operate on, ranging from flat text chunks to structured graphs, document-level units, and sequentially constructed evidence chains. Such differences reflect distinct design assumptions about how evidence should be retrieved and organized. These choices have direct implications for reasoning capability, system complexity, and deployment cost.

4.2.1 Basic RAG

Basic RAG is the canonical formulation of RAG, in which a retrieval component is combined with a generative language model (e.g., T5 or BART) to incorporate external knowledge at inference time. In this pipeline, knowledge sources are represented as a flat collection of text chunks, which serve as the basic units for retrieval and evidence selection [68]. This design enables models to condition generation on explicitly retrieved information, a requirement for knowledge-intensive tasks where relevant evidence cannot be reliably stored within model parameters alone.

In this flat representation, retrieval methods select a fixed top-K set of passages relevant to a given query, typically employing either lexical or dense semantic matching [69]. Although these approaches effectively identify relevant content, they function at the passage level and do not explicitly model relationships among retrieved chunks. A re-ranking stage is often applied to refine the retrieved candidates, which improves precision but introduces additional computational overhead [69,70].

The appeal of Basic RAG lies in its ability to decouple knowledge from model parameters, allowing updates to external knowledge sources without retraining the generator. Nevertheless, its flat, chunk-based representation presents a significant limitation by treating retrieved passages as independent units. This approach impedes the capture of cross-passage relationships and higher-level document structure [69]. Consequently, relevant evidence may become fragmented, and the model may encounter difficulties with queries that demand multi-step reasoning or structured comprehension. These challenges highlight the structural constraints inherent in pipeline-based RAG and underscore the need for more advanced architectures.

4.2.2 GraphRAG

GraphRAG extends the conventional RAG paradigm by replacing flat text chunks with graph-structured evidence representations. Instead of treating retrieved passages as independent units, GraphRAG organizes evidence as nodes and encodes semantic, logical, or referential relationships as edges. This method is motivated by the observation that, for complex queries, relevant information is often distributed across multiple related textual units, making isolated passage retrieval insufficient for reliable evidence aggregation [71,72]. As illustrated in Fig. 6, while Basic RAG relies on disconnected text fragments, GraphRAG transforms them into a structured knowledge network via entity extraction and explicit relationship linking, enabling more coherent evidence organization.

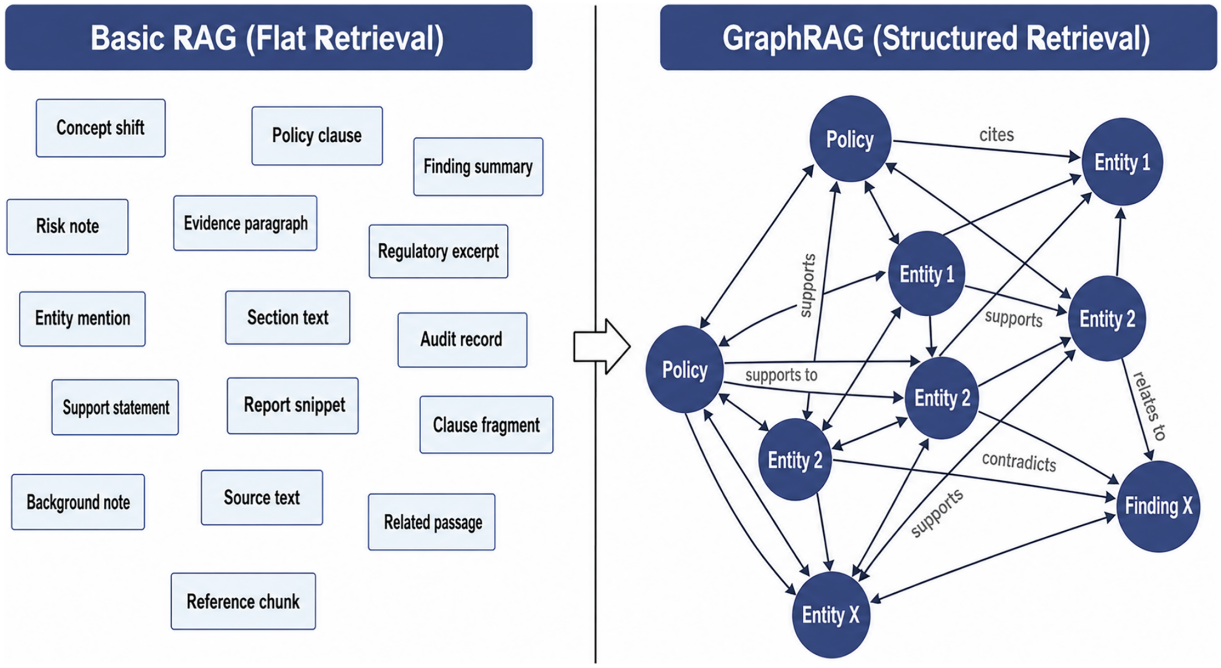

Figure 6: Comparison of knowledge organization between basic RAG and GraphRAG.

In GraphRAG pipelines, retrieval operates over structured representations where evidence relevance is determined by both local similarity and relational context. This structure facilitates more coherent evidence aggregation, particularly when no single text unit contains sufficient information to answer a query [71–73].

This approach is particularly effective for multi-hop reasoning and for domains with strong internal dependencies, where relevant evidence spans multiple linked units. By preserving relational structure, GraphRAG reduces the need for the language model to infer connections from loosely associated fragments and improves coverage of indirectly related information [72,74].

However, these advantages involve trade-offs. Constructing and maintaining graph-structured representations increases preprocessing complexity, and structured retrieval generally incurs higher computational costs. More significantly, GraphRAG is fundamentally a pipeline-based design in which retrieval and reasoning follow predefined procedures. Consequently, it lacks the flexibility to dynamically adapt to unexpected reasoning failures or to decompose complex tasks beyond predefined structures. This limitation underscores the boundaries of structured RAG and motivates the development of more adaptive and controllable systems.

4.2.3 FileRAG

FileRAG addresses a fundamental limitation of flat chunk-based retrieval in long and structured documents. In practical applications, relevant information is frequently distributed across contiguous sections that collectively establish meaning. Partitioning these documents into fixed-length chunks often disrupts contextual continuity, resulting in evidence that is locally relevant but globally incomplete. To mitigate this issue, FileRAG redefines the retrieval unit from isolated passages to document-centric structures, such as entire files or semantically bounded sections.

The necessity of maintaining document-level coherence is evident in tasks where answers span entire document regions rather than isolated passages, as demonstrated by datasets such as Natural Questions [75].

By preserving document boundaries, FileRAG mitigates a common failure mode of flat RAG pipelines in long-context scenarios, where top-K passage retrieval requires the model to reconstruct document structure from dispersed fragments. Supplying structurally consistent evidence enhances robustness when positional and sectional relationships are critical [75,76].

These advantages, however, introduce new trade-offs. Document-level retrieval increases input length and places greater demands on the model's context window, resulting in higher computational costs [76]. Long-document architectures such as Longformer mitigate this issue through sparse attention mechanisms, but they do not fully resolve the challenges of reasoning and attribution when integrating long-context evidence [77]. Additionally, coarse retrieval granularity amplifies the consequences of retrieval errors, as irrelevant context may dominate the input. Furthermore, FileRAG retains a pipeline-based architecture in which retrieval units are predefined and static. Consequently, it lacks the flexibility to adjust retrieval strategies according to query complexity or intermediate reasoning outcomes, which limits its effectiveness in addressing diverse or dynamically evolving tasks.

4.2.4 Multi-Hop RAG

In a Multi-hop RAG pipeline, retrieval and generation processes are interleaved across multiple steps. Rather than retrieving a fixed set of passages once, the system iteratively conditions each subsequent retrieval on intermediate outputs or partial evidence. This sequential retrieval paradigm externalizes aspects of the reasoning process within the retrieval mechanism, facilitating the discovery of information not directly accessible through the initial query [78,79].

As shown in Fig. 7, the process begins with an initial query and a first retrieval step (Hop 1), followed by intermediate reasoning that refines the information need and directs subsequent retrieval (Hop 2). This iterative accumulation of evidence enables the system to systematically expand the evidence space.

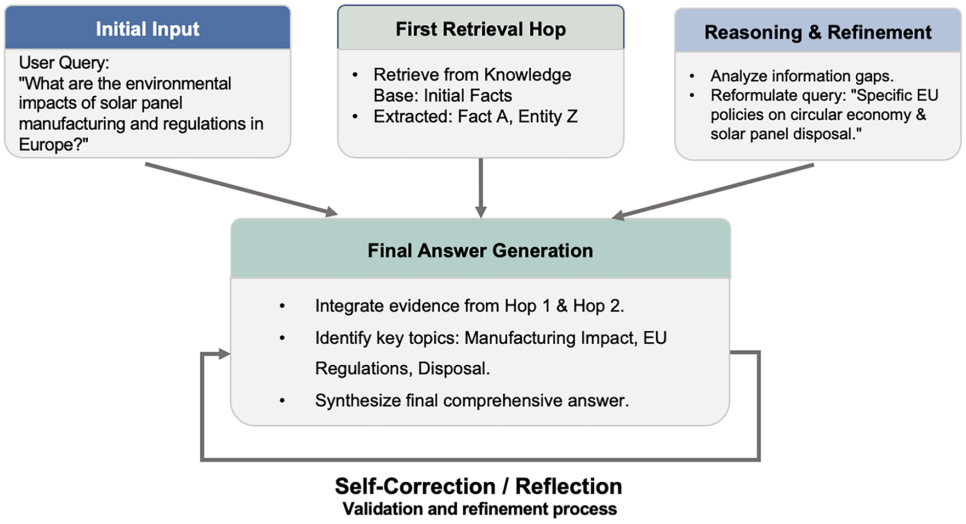

Figure 7: Multi-hop RAG sequential flow.

Multi-hop RAG is particularly effective for tasks requiring reasoning over distributed or implicit evidence. By refining retrieval based on intermediate reasoning, it mitigates a key limitation of flat RAG pipelines, where early retrieval decisions constrain downstream generation [78].

However, these advantages come with trade-offs. Iterative retrieval increases latency and computational cost, and errors introduced at early stages may propagate and affect subsequent retrieval steps [79]. More importantly, although Multi-hop RAG introduces a tighter coupling between retrieval and reasoning, the process still follows predefined execution patterns. As a result, it lacks the ability to dynamically adjust reasoning strategies or control the retrieval process in response to evolving task requirements, underscoring the need for more adaptive and controllable systems.

4.2.5 Operational Considerations: Chunking, Update Synchronization, and Serving Efficiency

In addition to pipeline design, practical RAG systems are constrained by operational factors including chunking, update synchronization, and serving efficiency. These factors have a direct impact on retrieval reliability and system behavior across all system variants.

As summarized in Table 4, chunking strategies require balancing efficiency, contextual coherence, and adaptability. Furthermore, language models do not consistently leverage extended contexts, and increasing input length does not necessarily enhance generation quality [80,81].

Table 4: Operational trade-offs of chunking strategies in RAG systems.

Strategy	Granularity Basis	Advantages	Limitations
Fixed size	Token/length	Simple, efficient	Context fragmentation
Structure-aware	Section/layout	Semantic coherence	Preprocessing cost
Dynamic	Query-aware	Adaptive relevance	Increased coordination and serving complexity

Retrieval quality is further affected by inconsistencies introduced during asynchronous updates to the indexing and embedding components, which may result in evidence mismatches in downstream stages. In addition, serving constraints, particularly latency requirements, often force systems to trade completeness for responsiveness, resulting in imperfect or partial evidence [82,83].

In summary, the discussed RAG variants significantly improve the system's ability to manage complex data structures and multi-step retrieval. However, these systems remain largely static and deterministic, following predefined execution paths without the flexibility to address unexpected reasoning failures or autonomously decompose tasks. This limitation of fixed pipelines highlights the need for Agentic RAG systems, which introduce dynamic decision-making and autonomous control throughout the generation lifecycle, as discussed in the next section.

4.3 Agentic RAG: From Query Assistance to Task-Oriented Problem Solving

While earlier RAG variants improve retrieval and contextual grounding, they are still limited by single-pass pipeline execution. In many real-world tasks, intermediate results change subsequent information needs, making such static setups inadequate. This limitation drives the development of agentic RAG, which allows for iterative, task-focused control over retrieval and generation.

4.3.1 Limitations of Pipeline-Centric RAG Variants and the Need for Agentic Control

Pipeline-centric RAG architectures, such as Basic RAG, GraphRAG, FileRAG, and Multi-hop RAG, address specific limitations in retrieval quality, evidence organization, and reasoning depth. However, recent work on adaptive retrieval-augmented generation shows that fixed retrieval behaviors are insufficient for

diverse queries because systems may need to dynamically determine when retrieval is necessary and how to coordinate it during inference [84].

This rigidity poses challenges in complex tasks where intermediate results influence subsequent information requirements. For instance, in regulatory analysis or contract review, newly identified constraints can invalidate previous assumptions and necessitate revisiting earlier steps. Recent research on Agentic RAG emphasizes that complex tasks require dynamic coordination among retrieval, reasoning, tool use, and evaluation, rather than a single-pass retrieval-generation pipeline [85]. Pipeline-based RAG systems do not provide explicit mechanisms for task decomposition, conditional branching, or self-correction, which complicates error management as task complexity increases.

Agentic RAG mitigates these limitations by introducing an explicit control layer that orchestrates retrieval and generation processes over time. Rather than treating retrieval as a single preparatory step, agentic systems decompose user objectives into intermediate subgoals, dynamically select appropriate tools, and revise prior decisions when inconsistencies are detected. This approach is consistent with recent reasoning-action frameworks, in which language models interleave reasoning with external tool use to enable adaptive decision-making [86,87]. Through the integration of planning, retrieval, and reflection, agentic RAG enables retrieval-augmented generation to function as an adaptive process suitable for real-world, multi-step tasks.

4.3.2 Representative Agentic Frameworks and Tooling Ecosystems

To operationalize agentic RAG, several frameworks have been proposed to support planning, tool invocation, and iterative control during task execution. These systems advance beyond fixed pipelines by introducing abstractions for state management, action coordination, and revision of intermediate decisions.

LangGraph models agent behavior as a stateful execution graph in which nodes represent reasoning or tool steps and edges define control flow, enabling explicit tracking of task structure and intermediate states [88]. AutoGen extends this paradigm through multi-agent collaboration, allowing specialized agents to coordinate via structured communication and enhancing robustness through role-based decomposition and cross-checking [89].

In parallel, tool-centric reasoning frameworks offer a complementary foundation. ReAct introduces a reasoning-action loop that interleaves natural-language reasoning with tool use, while Toolformer demonstrates that models can learn when to invoke tools using self-supervised signals [86,87].

Collectively, these frameworks illustrate a shift in RAG system design from optimizing individual components to orchestrating interactions among reasoning, tools, and generation. By providing explicit mechanisms for state management, role specialization, and dynamic control, agentic tooling ecosystems enable RAG systems to scale beyond query assistance toward sustained, task-oriented problem solving.

4.3.3 System-Level Synthesis: From Pipeline RAG to Agentic RAG

The RAG paradigms discussed in Sections 4.1–4.3 should be understood as complementary responses to distinct system-level constraints rather than as competing alternatives. Although earlier variants differ in retrieval structure and reasoning capability, the primary distinction concerns the allocation of control over task execution within the system.

Pipeline-centric RAG incrementally enhances performance by optimizing individual components of the retrieval-generation process. Basic RAG grounds generation in external evidence, whereas later variants

improve relational structure, contextual coherence, and reasoning depth. Nevertheless, these designs generally assume a fixed execution flow, which restricts adaptability when intermediate results alter subsequent information requirements.

Agentic approaches overcome this limitation by introducing explicit control over task execution. Rather than relying on fixed pipelines, these methods enable dynamic coordination among reasoning, retrieval, and tool use, allowing intermediate decisions to inform subsequent actions [86,90]. This transition marks a shift from static evidence grounding to adaptive, decision-driven execution.

To clarify this progression, Table 5 presents representative RAG paradigms from a system-level perspective, highlighting their functional roles, degree of execution control, and typical task scope. This comparison explicitly demonstrates how RAG evolves from predefined pipelines to systems capable of dynamic task orchestration. Within this framework, Agentic RAG constitutes a qualitative architectural shift rather than a mere incremental extension of retrieval. Retrieval and generation are conceptualized as callable operations, with their use determined during execution. This approach enables systems to revise intermediate decisions and adapt to evolving task states. Recent system-level analyses further substantiate this transition toward orchestrated, multi-stage workflows that integrate reasoning, tools, and intermediate-state management [91].

Table 5: System-level comparison of RAG paradigms by design motivation and functional role.

RAG Paradigm	Core Technical Mechanism	Primary Limitation Addressed	Suitable Task Scope	Role in the Overall System
Basic RAG	Single-step retrieval and generation with minimal control logic	Lack of access to external, updatable evidence in parametric models	Simple factual queries, FAQs	Lightweight knowledge grounding layer
GraphRAG	Relation-aware evidence linking and structured reasoning	Inability of flat retrieval to capture dependencies and relational constraints	Policies, regulations, rule comparison	Relational reasoning enhancement layer
FileRAG	Document-centric retrieval and long-context aggregation	Context fragmentation and loss of cross-section coherence caused by fine-grained chunking	Contracts, reports, manuals	Context integrity preservation layer
Multi-hop RAG	Stepwise retrieval and evidence accumulation with fixed execution flow	Insufficient reasoning depth and limited adaptability in single-round retrieval	Analytical and synthesis tasks	Reasoning depth extension layer
Agentic RAG	Task planning, dynamic tool orchestration, and reflective control	Lack of adaptability, error correction, and decision-level control in pipeline-based systems	Multi-step, high-risk enterprise tasks	Integration and decision control layer

Although agentic RAG offers increased flexibility and adaptability at the task level, it introduces system-level risks that are not present in pipeline-centric architectures. First, multi-step planning and dynamic tool

invocation can result in unpredictable computational costs, as execution paths are variable and may expand based on intermediate decisions. Second, tool usage generates more complex error modes, where incorrect intermediate outputs, faulty tool calls, or misaligned retrieval results may propagate across steps and accumulate over time. Prior research on language agents demonstrates that errors in intermediate reasoning or feedback signals can substantially affect subsequent decisions, resulting in compounded deviations in multi-step scenarios [92]. Third, validating the correctness of agentic behavior becomes considerably more difficult, since the reliability of the final output depends on both individual steps and the overall consistency and coherence of the multi-step reasoning process. Existing studies indicate that large language models continue to face challenges in planning and evaluating multi-step tasks due to the absence of robust intermediate verification mechanisms [93].

These risks collectively shift the focus from improving individual step accuracy to ensuring system-wide controllability, where cost, error propagation, and planning uncertainty must be managed together. In this work, such risks are addressed through a risk-controlled system design that integrates feedback-driven data flywheel mechanisms (Section 5) with agentic workflow control and guardrail-based validation (Section 7), enabling both adaptive execution and bounded reasoning behavior.

5 Evaluation and Zero-Hallucination Metrics

In enterprise deployments, document understanding, and RAG are widely adopted to incorporate external evidence and reduce hallucination risks in large language models. However, the retrieval of evidence alone does not guarantee factual faithfulness. Even when relevant documents are accessible, models may generate unsupported inferences or incorrectly synthesize information from multiple sources. Consequently, the central concern shifts from the mere presence of citations to whether those citations substantiate the underlying claims. A response may appear semantically coherent while remaining inconsistent with its cited sources. Thus, attribution and faithfulness are not solely aspects of generation quality but are fundamentally issues of auditability and verifiability [94].

In this context, Zero-Hallucination should be regarded not as the elimination of all errors, but as a guiding design principle. By employing claim decomposition, reasoning constraints, and verification and repair mechanisms, errors can be detected, localized, and systematically managed. Evaluation therefore extends beyond model comparison and becomes integral to the deployment process. In enterprise environments, evaluation is an ongoing process rather than a single validation step. Outputs that satisfy attribution and faithfulness criteria should inform subsequent retrieval strategies, prompt configurations, and knowledge base updates, thereby establishing a closed loop of data, generation, evaluation, and optimization. This feedback loop supports the data flywheel approach in enterprise-grade RAG systems by converting evidence, evaluation results, and risk signals into reusable optimization assets. It prioritizes structured evaluation and feedback to drive ongoing improvements in retrieval accuracy, generation quality, and governance, shifting evaluation from passive detection to active control [95].

To address the reliability of RAG systems and to integrate evaluation within this closed-loop structure, this section organizes existing research into a three-level evaluation framework:

(1) Claim-level attribution and faithfulness verification:

Generated outputs are decomposed into atomic claims and aligned with supporting evidence (e.g., AIS and FActScore), thereby establishing fine-grained traceability.

(2) System-level automated and scalable evaluation mechanisms:

Approaches such as model-as-a-judge, reference-free RAG diagnostics, and engineered semantic scoring enable evaluation to function within continuous monitoring workflows.

(3) Governance-level benchmark datasets and stress testing:

Document-understanding benchmarks, high-risk domain datasets, and institution-aligned evaluations translate compliance and auditing requirements into verifiable conditions.

This three-level structure aims to advance from outputs that only appear correct to reliability that is measurable, traceable, and continuously improvable, thereby supporting long-term enterprise deployment and auditing of generative systems.

5.1 Attribution & Faithfulness

In enterprise RAG systems, attribution and faithfulness are essential for ensuring traceability and compliance, particularly when generated content must be grounded in verifiable evidence. To operationalize these requirements, attribution must be embedded within the generation process rather than treated as a post-hoc evaluation step.

Fig. 8 presents a claim-level attribution and repair pipeline designed to convert raw model outputs into auditable responses.

(1) Raw Output: The original response generated by the LLM (unverified), which may contain multiple factual claims, potential hallucinations, and no explicit evidence alignment.
(2) Claim Decomposition (FActScore): Long-form text is decomposed into atomic claims, converting a holistic response into independently verifiable units. This step enables precise identification of unsupported statements.
(3) Atomic Claims: A set of decomposed individual claims.
(4) Evidence Retrieval: For each atomic claim, a retrieval query is issued to obtain candidate evidence passages.
(5) Claim-Evidence Pairs: Each claim is paired with retrieved evidence, forming the basic verification unit.
(6) NLI-based Confidence Scoring: A Natural Language Inference (NLI) model determines whether the evidence entails the claim.
(7) Attributed Claim: Claims that meet the confidence threshold are treated as evidence-supported and can be included in the final response.
(8) Revision Based on Retrofit Attribution Using Research and Revision (RARR): Claims that fail to meet the threshold trigger a repair process, which may involve additional retrieval, revision of the original text, or removal of unsupported content.
(9) Auditable Response: The final response integrates all supported claims and revised content.

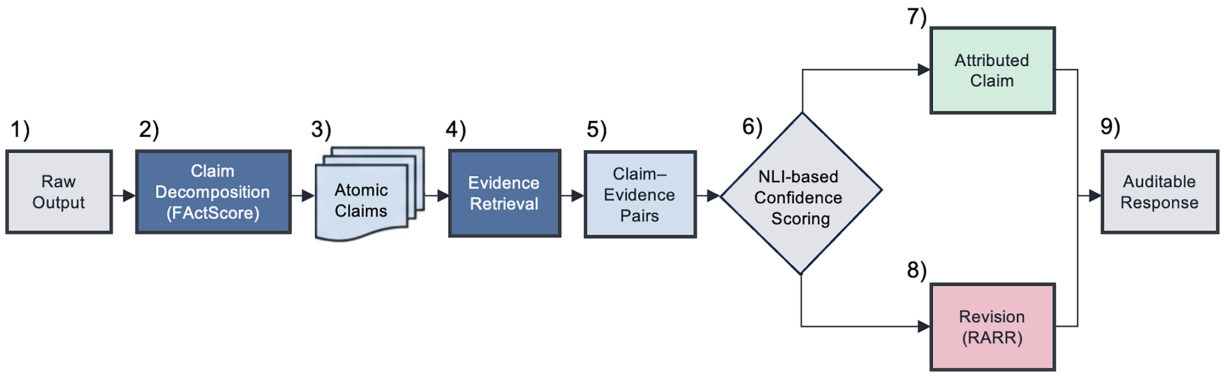

Figure 8: Attribution pipeline: from output to atomic claims, evidence matching, confidence scoring, and patching.

The primary advantage of this design is the integration of attribution within the generation workflow, rather than limiting it to an evaluation metric. Acceptance of generated content depends on its fulfillment of structured evidence verification, rather than fluency or surface plausibility. This multi-stage process constrains probabilistic language generation through explicit evidence alignment, thereby enhancing auditability and traceability in enterprise RAG systems. Building on this foundation, existing research is organized into a three-level framework for attribution and faithfulness, as summarized in Fig. 9. The first level, Claim-Level Attribution, involves decomposing generated outputs into independently verifiable atomic claims and assessing whether each claim is supported by retrieved evidence using methods such as AIS and FActScore. This level shifts the focus from evaluating the overall reasonableness of a response to determining whether each individual claim is evidence-supported, thereby establishing a fine-grained attribution baseline [94,96].

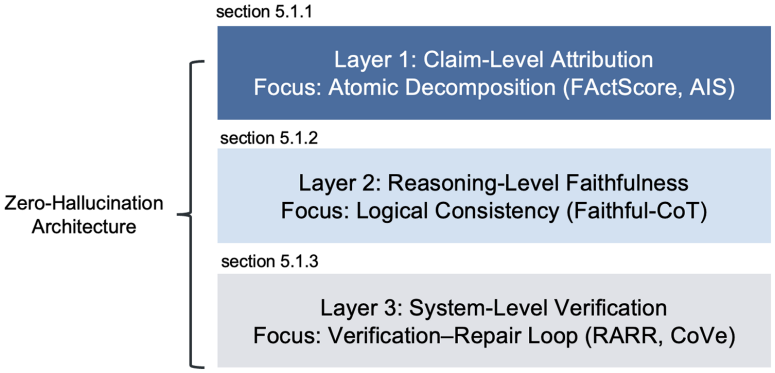

Figure 9: Three-level attribution and faithfulness framework for zero-hallucination governance in enterprise RAG systems.

The second level, Reasoning-Level Faithfulness, ensures logical consistency between intermediate reasoning steps and final conclusions. Instead of evaluating only final outputs, approaches such as Faithful-CoT impose logical constraints on reasoning chains, thereby reducing the risk of introducing unsupported assumptions during multi-step inference [97].

The third level, System-Level Verification, implements a verification and repair loop (e.g., CoVe, RARR), focusing on post-generation validation and correction to address unsupported or contradictory content without retraining the base model. Thus, attribution becomes a continuous deployment mechanism rather than a one-time assessment [98,99].

Overall, the three-level structure in Fig. 9 frames attribution as a layered design spanning claim verification, reasoning consistency, and system-level repair, collectively enhancing auditability and traceability in high-risk settings.

5.1.1 Claim-Level Attribution

A common limitation of current generative systems is that a single response frequently includes multiple factual statements. While some statements are directly supported by retrieved evidence, others result from statistical inference or the model's implicit completion. When evaluation treats the entire response as a single unit, it becomes difficult to distinguish between wholly unsupported content and partially supported claims. This coarse-grained assessment conceals fine-grained attribution failures and restricts precise error localization, making it difficult to trace responsibility or revise specific problematic statements in enterprise contexts.

The Attributable to Identified Sources (AIS) framework formalizes attribution in terms of whether generated content is supported by specified source documents, rather than merely being consistent with general world knowledge. A primary contribution of AIS is the operationalization of attribution as a testable condition: for each generated statement, it must be possible to reasonably assert that "according to the source document, this claim holds." Each statement must also be independently interpretable and verifiable within its context. By defining attribution in relation to identified sources, AIS distinguishes factual correctness from source traceability. This distinction is critical in enterprise applications. Even when a statement is factually accurate, the inability to trace it to approved knowledge sources constitutes attribution failure. This formalization renders attribution a definable, evaluable, and reproducible technical criterion, rather than a loosely interpreted citation practice [94].

However, in long-form generation tasks, sentence-level evaluation is insufficient for detecting fine-grained attribution deviations. A single sentence may contain multiple separable informational components, some of which are supported by evidence while others are not. As a result, binary sentence-level labeling obscures cases of partial attribution failure. Factual Precision in Atomicity Score (FActScore) addresses this limitation by decomposing generated text into atomic facts and verifying each atomic unit against designated sources. This atomic-level approach enhances error localization and enables attribution performance to be quantified as a continuous metric, rather than as a binary judgment at the response level. Notably, FActScore defines factuality in relation to specific knowledge sources, rather than presuming a universal ground truth. This property enables natural alignment with enterprise knowledge bases, regulatory documents, and version-controlled repositories, making FActScore suitable for deployment-oriented evaluation scenarios [96].

If outputs are not broken down into verifiable claims, attribution remains coarse-grained and cannot clearly separate unsupported from partially supported content. AIS defines attribution through source grounding, while FActScore builds on this by using atomic decomposition and proportional measurement. Together, these methods make attribution a precise, measurable diagnostic process and provide a basis for reasoning and system-level faithfulness control in enterprise RAG systems.

5.1.2 From Claim-Level Attribution to Reasoning-Level Faithfulness

Atomic claim-level attribution aligns individual statements with identifiable evidence, yet such alignment does not guarantee overall decision reliability. In enterprise settings such as regulatory comparison, financial analysis, or medical decision support, outputs rely on multi-step reasoning rather than isolated claims. Even when each statement is evidence-supported, the reasoning chain may contain logical gaps, implicit transitions, or incorrect deductions, leading the conclusion to diverge from the underlying evidence. The problem, therefore, extends beyond sentence-level factual accuracy. A response may contain no explicit errors and still produce a conclusion not logically entailed by its own reasoning. Coherent Chain-of-Thought explanations can function as post-hoc rationalizations, where structured intermediate steps fail to justify the answer. This inconsistency between the reasoning and the conclusion constitutes a reasoning-level hallucination.

Faithful Chain-of-Thought addresses this gap by distinguishing linguistic expression from logical execution. This method transforms a problem into a structured or symbolic representation, followed by the application of a deterministic solver to perform the required computation or inference. As a result, the reasoning process becomes transparent and inspectable, rather than remaining embedded in free-form text. Errors can be localized to either the translation or solving stage, rather than being concealed within narrative explanations. Empirical results demonstrate that this separation enhances the consistency of reasoning and reduces divergence between intermediate steps and final answers in structured reasoning tasks [97].

However, this mechanism is not universally applicable, as it presupposes that those problems can be formalized in a structured manner. In open-domain or semantically ambiguous contexts, reasoning cannot always be reduced to deterministic procedures. Furthermore, if implicit assumptions are not explicitly stated, inaccuracies may be introduced during symbolic translation. These limitations indicate that reasoning-level faithfulness can mitigate logical risk but cannot fully eliminate it.

Extending attribution from atomic claims to reasoning structure shifts the focus from verifying the existence of supporting evidence to evaluating the validity of inference. The former addresses source support, while the latter pertains to inferential consistency. Collectively, these criteria establish complementary conditions for reliability in enterprise RAG systems.

5.1.3 Verification-Repair Loop Mechanism

Although claim-level attribution and reasoning-level alignment enhance the verifiability of generated content, previous research indicates that these mechanisms are insufficient in deployment scenarios involving extended contexts and iterative interactions. Errors in large language models are dynamic and cumulative. Even when individual claims are supported by evidence and reasoning appears internally consistent, localized misinterpretations may remain undetected and propagate through subsequent responses. This error propagation is particularly evident in long-form generation and cross-document integration tasks, where minor early deviations can influence later conclusions. In enterprise contexts such as legal analysis or financial reporting, this propagation often manifests not as explicit factual errors but as a gradual divergence from the intended evidence base. These findings indicate that static verification alone is inadequate for ensuring system-level reliability.

To address this limitation, Chain-of-Verification (CoVe) implements a post-generation verification stage that functions independently from the initial reasoning process. Following the generation of a preliminary response, the model generates verification questions, answers them in relatively isolated conditions, and then revises the original output based on these verification results. By decoupling generation from verification, CoVe reduces the risk that earlier reasoning errors are reinforced during correction. Empirical evaluations show that this factorized generate-verify-revise framework significantly reduces factual error rates in generated text [96]. Notably, this approach reconceptualizes faithfulness as an iterative verification process rather than a static evaluation criterion.

However, CoVe is primarily limited to verification at the time of generation. In operational settings, enterprise systems frequently require the re-examination of previously generated or legacy content that was not produced with structured attribution. RARR expands the verification paradigm by introducing a model-agnostic, post-hoc repair strategy. Instead of retraining the base model, RARR establishes an external research and revision pipeline. This pipeline automatically generates retrieval queries for identified claims, detects conflicts or missing support between claims and evidence, and revises the text while maintaining its original structure and tone. As a result, previously non-attributable outputs can be converted into traceable versions without altering the underlying generation model [99].

These approaches extend verification from claim-level checking to system-level repair, shifting the control of faithfulness from a one-time assessment to continuous correction. In this framework, zero hallucination is a design goal where errors are detectable, localizable, and correctable through layered verification and repair. This closed-loop structure supports auditability and traceability in enterprise RAG systems under dynamic conditions.

5.2 Automatic Evaluation for RAG and Summarization

In enterprise RAG and generative AI deployments, automated evaluation must go beyond offline benchmarking. It should run alongside the runtime system as part of governance. As shown in Fig. 10, the architecture uses an execution-evaluation-optimization data flywheel. During operation, the system continuously collects user queries, retrieves evidence, generates outputs, and produces evaluation results, integrating evaluation into the operational pipeline. This architecture can be organized into three core components [100,101].

(1) The Runtime System: processes user queries and generates responses with the relevant retrieval context.
(2) RAG Quality & Risk Diagnostics. Evaluates outputs across dimensions such as semantic quality, attribution faithfulness, and reference-free risk.
(3) Governance & Continuous Validation. Uses diagnostic results to inform system optimization, including prompt refinement, knowledge-based updates, and retrieval strategy adjustments.

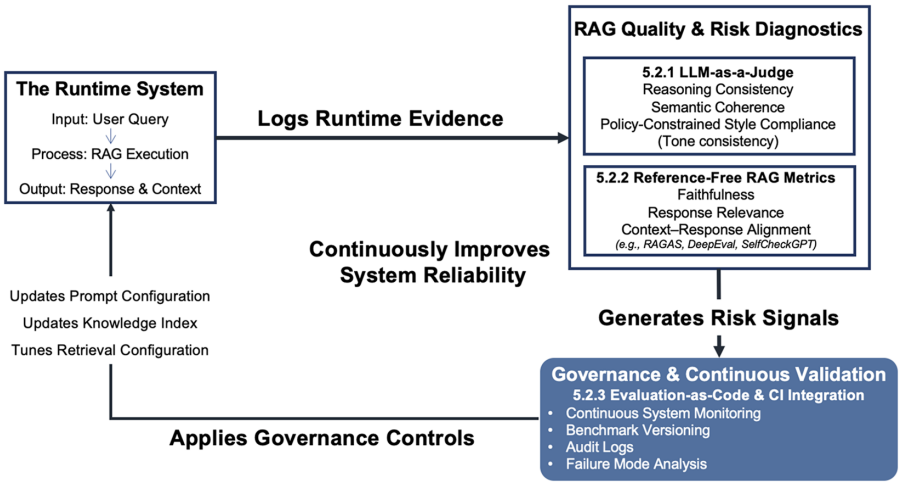

Figure 10: Data flywheel architecture for enterprise RAG governance.

With this design, evaluation becomes a continuous improvement mechanism rather than a one-time verification step. Each user interaction generates evaluable signals, each evaluation outcome informs optimization, and each optimization decision shapes future system behavior. This closed-loop approach makes automatic evaluation a core driver of enterprise RAG governance, not just an auxiliary testing procedure.

5.2.1 LLM-as-a-Judge

In enterprise RAG deployments, evaluation challenges stem less from generation quality than from the limitations of traditional metrics. Reference-based indicators such as BLEU and ROUGE rely on the assumption of n-gram overlap, yet prior studies demonstrate limited correlation with human semantic judgment and insufficient robustness in semantically diverse tasks. In domains such as contract interpretation, financial analysis, and knowledge summarization, multiple valid expressions may exist for correct answers. Overlap-based metrics are therefore inadequate for distinguishing semantically accurate paraphrases or identifying reasoning-level deficiencies [102,103].

Embedding-based approaches, such as BERTScore, enhance semantic alignment but function solely as offline metrics. These methods do not support open-ended evaluation in the absence of gold references, assess reasoning validity beyond surface similarity, or enable continuous monitoring amid dynamic updates.

To address these limitations, large language models have been employed as semantic evaluators through structured prompting, establishing the LLM-as-a-Judge paradigm. Empirical studies indicate strong concordance between advanced models and human experts across a range of generation tasks [104,105]. Notably, this paradigm formalizes semantic judgment as a systematic and repeatable reasoning process. In practice, this paradigm addresses subtle inconsistencies and misleading paraphrases in high-risk domains such as financial compliance and legal analysis. It enables the detection of citation inconsistencies, reasoning gaps, and unsupported summaries.

However, evaluation results may shift as task distributions evolve, and judge models are sensitive to prompt design and configuration parameters. Consequently, LLM-as-a-Judge should be regarded as an initial semantic quality indicator within the data flywheel, not as a definitive authority [106]. In this framework, runtime outputs are continuously evaluated and versioned, producing traceable quality signals that guide prompt refinement and retrieval adjustments. This generation, evaluation, and optimization cycle integrates semantic assessment into ongoing governance processes rather than treating it as a singular experiment [107].

Although effective for assessing semantic plausibility, LLM-as-a-Judge does not verify grounding in retrieved evidence and therefore constitutes only the semantic monitoring layer within the flywheel.

5.2.2 Reference-Free RAG Metrics

A key governance challenge in enterprise RAG systems is evaluating whether generated content is grounded in retrievable evidence when no gold-standard answer is available. Traditional reference-based metrics like ROUGE and BLEU correlate poorly with human judgment in open-ended tasks and do not capture semantic consistency or factual faithfulness. Importantly, factual correctness does not guarantee source-level attribution; even accurate content is an attribution failure if it cannot be supported by the designated evidence [94].

Enterprise deployment presents three main constraints: many internal QA and document analysis tasks lack canonical answers, organizations must separate retrieval errors from generation errors, and black-box settings limit access to reasoning traces and internal model signals. These factors make reference-free evaluation a necessary governance tool rather than a research convenience.

RAGAS addresses this need by breaking down RAG quality into three diagnostic dimensions [108]: faithfulness, answer relevance, and context relevance. This approach helps teams pinpoint failures in retrieval or generation and use evaluation outputs as structured feedback within the data flywheel. In practice, this is particularly valuable in enterprise settings where no ground-truth answers exist, such as internal document QA or compliance analysis. It enables systems to identify retrieval failures, incomplete context coverage, or hallucination risks without relying on labeled data. This improves continuous monitoring and error diagnosis in real-world deployments.

In black-box environments, SelfCheckGPT estimates hallucination risk by using repeated stochastic sampling and semantic consistency comparison, without relying on references or external knowledge bases [109]. Its purpose is to provide operational risk signals when direct attribution checks are not possible, rather than to certify correctness.

However, reference-free methods can introduce circular bias when models validate themselves and increase computational costs due to repeated sampling. These methods should be integrated with threshold policies and continuous monitoring to balance efficiency and reliability. The next subsection explains how to operationalize these metrics as part of a sustained governance infrastructure.

5.2.3 Evaluation-as-Code & CI Integration

In enterprise RAG systems, models and knowledge bases are dynamic rather than static assets. Documents are updated, indices are rebuilt, prompts are revised, and foundation models may be versioned. Under these conditions, systems that previously passed attribution and faithfulness checks may later demonstrate reduced attribution completeness, retrieval-generation misalignment, or metric drift. Consequently, reliability cannot be ensured through a single validation event. In the absence of ongoing audit and monitoring mechanisms, an accountability gap may develop between model behavior and organizational responsibility [110].

Continuous monitoring and traceable logging are therefore essential for effective governance. If attribution and faithfulness cannot be consistently measured and versioned, their regulatory significance diminishes. RAGChecker further supports engineering-oriented RAG evaluation by decomposing system quality into fine-grained diagnostic metrics for the retrieval and generation modules [111]. In practice, this approach reduces regression risks in high-stakes domains (e.g., finance and legal compliance) by detecting policy violations, outdated references, and retrieval misalignment before deployment. It enables traceable and auditable quality control aligned with regulatory requirements.

Building on this foundation, engineering frameworks such as DeepEval integrate evaluation mechanisms into unit testing and CI/CD workflows, making quality checks for generated content routine before and after system updates. Attribution completeness, faithfulness thresholds, and compliance-related constraints are encoded as executable rules. When model versions are updated, knowledge bases are modified, or retrieval strategies are adjusted, semantic and attribution metrics can be automatically recalculated and compared to historical baselines.

Within the architecture depicted in Fig. 10, this mechanism links RAG Quality and Risk Diagnostics with Governance and Continuous Validation. Evaluation-as-Code converts LLM-as-a-Judge and reference-free metrics into executable test modules, allowing outputs generated by the Runtime System to be continuously measured, version-controlled, and incorporated into optimization decisions. In this configuration, evaluation is integrated into the operational governance cycle rather than remaining a separate experimental process.

However, engineering-oriented monitoring is limited by structural constraints. Judge models can introduce stability and bias issues; test datasets may not accurately reflect real-world usage; and evaluation metrics that are misaligned with regulatory or organizational risk frameworks may satisfy technical thresholds while failing to meet governance expectations. While Evaluation-as-Code addresses the engineering challenge of continuous measurement, its effectiveness ultimately depends on dataset design and alignment with compliance requirements. The following section, therefore, examines compliance-oriented datasets and stress-testing frameworks.

5.2.4 Data Flywheel in Enterprise RAG Systems: A Closed-Loop Framework for Continuous Evaluation and Optimization

In enterprise-grade RAG systems, the primary challenge is not solely model performance, but rather the maintenance of long-term reliability and controllability amid continuously evolving data, models, and task requirements. Traditional static evaluation methods are insufficient for capturing dynamic risks encountered during real-world deployment, including updates to retrieved content, modifications in model versions, and changes in usage scenarios. Consequently, evaluation must shift from offline validation to an embedded operational mechanism that is closely integrated with system decision-making and optimization processes.

Within this framework, the data flywheel functions as both a closed-loop architecture and a guiding system design principle, transforming evaluation into a governance capability.

To address the limitations of static evaluation in real-world deployment, the data flywheel offers a closed-loop operational mechanism and redefines the function of evaluation in enterprise-grade RAG systems. Through iterative execution and continuous feedback, the system accumulates reusable optimization knowledge, resulting in a self-improving capability over time. The data flywheel integrates data collection, evaluation, feedback transformation, and system optimization into a continuously operating, system-level process. This integration enables enterprise RAG systems to sustain stability, controllability, and long-term adaptability in dynamic environments. However, in the absence of effective stability control mechanisms, the data flywheel may experience feedback loop instability, which can amplify erroneous signals across iterations. Table 6 presents four interconnected core stages, detailing their functional roles, implementation mechanisms, and system-level impacts to illustrate the operation of the data flywheel.

Table 6: Data flywheel components for continuous RAG optimization.

Stage	Core Function	Key Mechanisms	System-Level Impact
Runtime System & Evidence Logging [99]	Execute RAG pipeline and collect runtime evidence during system operation	User queries, retrieval results, generated responses, context traces, evidence attribution logs	Establishes traceability and foundational data
RAG Quality & Risk Diagnostics (5.2.1–5.2.2) [108,109]	Transform runtime outputs into structured quality and risk signals	LLM-as-a-Judge, reference-free metrics (e.g., RAGAS, SelfCheckGPT), faithfulness, context-response alignment	Enables automated monitoring and risk signaling
Governance & Feedback Transformation (5.2.3) [112]	Convert evaluation outputs into structured, actionable feedback signals	Error taxonomy, Reflexion, evaluation-as-code pipelines, audit logging	Bridges evaluation to action via auditable loops
System Update & Governance Control (5.2.3) [99]	Apply feedback to system-level optimization and policy enforcement	Retriever/reranker tuning, prompt engineering, knowledge base updates, policy-layer adjustment, CI/CD integration	Ensures continuous improvement and reliability

In enterprise deployment contexts, the data flywheel transforms evaluation into a continuously operating control mechanism. The system initially collects execution evidence and performs evaluations to generate quality and risk signals, which are subsequently converted into structured feedback and used to update the system. This iterative closed-loop process enables RAG systems to maintain traceable, auditable, and continuously improving reliability in dynamic environments. Nevertheless, the data flywheel introduces

certain limitations and risks. If the evaluation model is biased or unstable, erroneous signals may be amplified through repeated iterations, reinforcing errors, and causing the system to diverge from its intended objectives. Additionally, while reference-free evaluation enhances scalability, it may not reliably detect subtle semantic errors or implicit hallucinations without external fact verification.

Moreover, the data flywheel relies heavily on continuous collection and processing of execution evidence, which introduces additional computational costs and system complexity, thereby imposing higher infrastructure requirements for enterprise deployment. Finally, excessive reliance on internal feedback mechanisms may cause the system to converge toward a specific distribution, reducing its adaptability to novel queries or changes in external knowledge. To further illustrate how these advantages and limitations manifest in practice, Table 7 systematically summarizes failure modes across different application scenarios, corresponding evaluation signals, feedback transformation methods (structured feedback), and the resulting system optimization strategies.

Table 7: Data flywheel-driven continuous optimization across enterprise RAG scenarios.

Scenario	Runtime Evidence & Failure Pattern (Logs Runtime Evidence)	Risk Signals (RAG Quality & Risk Diagnostics)	Governance & Feedback Transformation	System Update & Governance Control
Financial Compliance QA [113,114]	Incorrect or outdated regulatory references; attribution mismatch	Faithfulness, citation consistency, evidence attribution alignment	Error attribution (retrieval vs. reasoning); outdated document detection	Knowledge base update; retrieval weighting adjustment; policy-aware reranking
Enterprise Knowledge Assistant [112]	Incomplete evidence coverage; hallucinated synthesis; missing supporting documents	Context coverage, evidence completeness, hallucination risk indicators	Multi-hop retrieval gap identification; knowledge fragmentation analysis	Multi-stage retrieval enhancement; reranker tuning; knowledge graph augmentation

The application of the data flywheel in enterprise-grade RAG systems can be demonstrated through two representative scenarios, each highlighting its operation under varying task and risk conditions and its associated benefits and limitations.

(a) Case 1: Financial Compliance Question Answering (Regulatory QA)

This scenario involves runtime evidence, risk signal generation, and system optimization. Key failure modes include the use of incorrect or outdated regulatory references and attribution mismatches, both of which may directly result in compliance violations and legal risks.

By employing LLM-as-a-Judge and reference-free metrics, the system generates risk signals such as faithfulness and citation consistency, while distinguishing between retrieval failures and reasoning errors. These signals inform actions, including knowledge-based updates and policy-aware reranking, thereby reducing error accumulation and enhancing traceability. This closed-loop mechanism is essential in financial contexts, where errors may otherwise be repeatedly amplified. However, if the evaluation model exhibits

bias or insufficient domain understanding, error signals may be reinforced, thereby compromising decision reliability [115].

(b) Case 2: Enterprise Knowledge Assistant

This scenario demonstrates the role of the data flywheel in co-optimizing context coverage and the retrieval-generation process. Primary failure modes include insufficient evidence coverage, hallucinated synthesis, and inadequate supporting documentation, which can result in incomplete or biased decisions.

Risk signals, including context coverage and evidence completeness, facilitate the identification of multi-hop retrieval gaps and knowledge fragmentation, prompting optimizations such as multi-stage retrieval and adjustments to the reranked. However, excessive reliance on internal feedback may lead to convergence toward existing knowledge distributions, reducing adaptability and leading to knowledge stagnation. Consequently, external validation and data diversity are necessary [116].

Building on the above observations, the empirical cases discussed in this study can be abstracted into a unified closed-loop workflow, as illustrated in Fig. 11. Specifically, the process begins with user query ingestion and contextual formulation, followed by retrieval through the RAG execution pipeline. The generated responses are then evaluated via LLM-as-a-Judge mechanisms to assess relevance, faithfulness, and alignment, producing structured risk signals such as hallucinations, omissions, or low-confidence outputs. In summary, although the data flywheel enhances performance through continuous evaluation and feedback, it does not fully prevent bias amplification or error accumulation. External validation, such as human review, benchmarking, and third-party audits, along with governance strategies including threshold policies, benchmark versioning, and failure monitoring, remain essential. The integration of evaluation-as-code with continuous monitoring supports sustained optimization and ensures reliable, traceable system behavior.

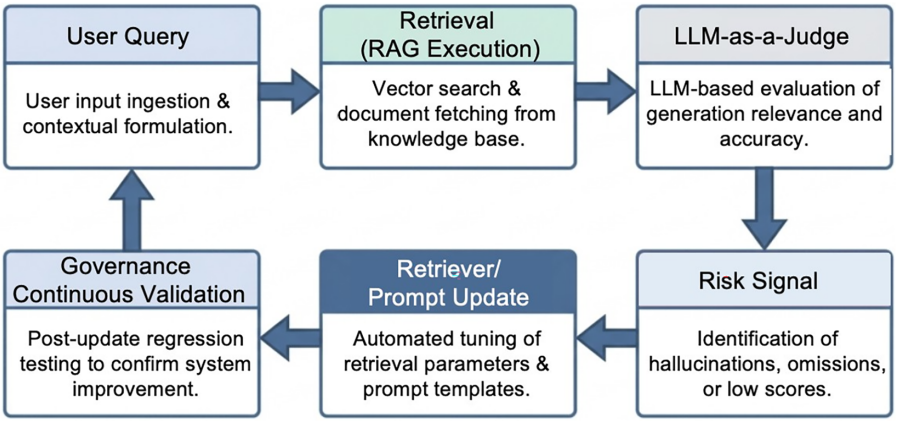

Figure 11: Data flywheel-based closed-loop workflow for continuous RAG optimization.

5.3 Benchmark Datasets and Compliance-Oriented Extensions

The reliability of enterprise RAG systems depends not only on verification mechanisms or evaluation procedures, but more importantly on whether benchmark datasets accurately reflect real deployment scenarios. If test corpora do not capture document complexity, domain-specific features, and compliance-related risks, even robust evaluation frameworks may not fully assess system performance.

This section reviews key benchmark datasets relevant to enterprise deployment, analyzing them across three complementary categories: document-structure understanding, high-risk-domain reasoning, and multimodal or evaluation-aligned benchmarks.

5.3.1 Document Structure Understanding Benchmarks

In enterprise RAG systems, layout displacement, column misalignment, or incorrect hierarchical interpretation during document parsing can result in distorted semantic representations for retrieval and generation. Even with correct attribution and faithfulness mechanisms, outputs may still show semantic drift or structural hallucinations. Document-level understanding is therefore essential in the evaluation framework.

Errors such as misreading multi-column layouts, misaligned table fields, or misplaced clause hierarchies can lead to amplified factual errors or attribution failures in downstream RAG processes. While document-level benchmarks evaluate whether models can interpret layout, visual features, and textual semantics, RAG-oriented benchmarks further evaluate whether retrieved evidence is accurately incorporated during generation. RAGTruth offers a hallucination corpus for retrieval-augmented generation and facilitates the assessment of factual inconsistencies and attribution failures in generated responses [117]. This approach tests structural-semantic integration rather than simple text recognition.

From a governance perspective, document structure benchmarks serve as the initial stress layer in a zero-hallucination framework. Distorted document parsing can cause attribution checks and automated diagnostics to misidentify risks by confusing structural errors with reasoning errors. In high-risk contexts such as legal contracts and financial reports, structural-semantic misalignment becomes a systemic risk. Document-level evaluation is therefore the foundation of compliance-oriented assessment.

5.3.2 Professional-Domain and High-Risk Reasoning Benchmarks

Enterprise AI systems are often used in high-risk areas such as legal review, financial analysis, and policy interpretation. In these contexts, errors can result in contractual misinterpretations, financial disclosure inconsistencies, or regulatory violations. Traditional evaluation metrics that focus on factual accuracy or fluency do not fully address these risks. In professional domains, models must demonstrate clause-level precision, numerical consistency, and verifiable reasoning. As a result, professional-domain benchmarks are essential for robust enterprise governance.

In the legal domain, CUAD (Contract Understanding Atticus Dataset) evaluates fine-grained clause recognition and semantic interpretation under legally binding contexts. Rather than focusing on surface extraction accuracy, it assesses whether models correctly identify obligations, responsibilities, and exception clauses whose legal effects may differ despite textual similarity. Through expert annotation and high-precision classification, CUAD enables legal-semantic errors to be explicitly quantified [118].

In financial contexts, FinQA focuses on multi-step numerical reasoning using financial reports, integrating both tabular data and text. The main risk is numerical inconsistency, as fluent responses may still breach compliance if calculations are incorrect. By including computational reasoning chains in its evaluation, FinQA moves assessment beyond semantic alignment to ensure verifiable numerical accuracy [119].

LegalBench evaluates legal reasoning across contract interpretation, case analysis, and regulatory application. Its multi-task design tests reasoning stability in different contexts, addressing deployment risks when performance declines under task changes [120].

These benchmarks reflect a clear shift from evaluating general factual correctness to assessing risk-sensitive, institution-aware reasoning. They provide a technical basis for measuring professional-domain reliability, yet their scope remains largely task-oriented and does not fully incorporate mechanisms for version traceability or regulatory alignment.

5.3.3 Multimodal and Evaluation-Alignment Benchmarks

In enterprise document systems, the principal risk associated with multimodal models arises from cross-modal misalignment rather than surface-level language errors. For example, when a model incorrectly associates chart titles with data axes, misaligns cross-column information, or fails to maintain layout dependencies, downstream RAG or reasoning modules may produce responses that appear coherent but are based on misaligned evidence. These errors are often challenging to identify using text-only evaluation frameworks, as the final output may remain fluent and internally consistent while lacking valid cross-modal grounding. This issue is referred to as cross-modal hallucination, in which incorrect visual-language correspondences distort the reasoning process at an early stage and propagate through subsequent stages of generation [121].

In addition to perception-level risks, multimodal evaluation introduces the challenge of aligning evaluation criteria with institutional standards. Even when technical performance metrics are satisfactory, automated evaluation may diverge from human or regulatory judgment if the scoring criteria are not explicitly aligned with domain-specific requirements. G-Eval addresses this challenge by providing a structured evaluation framework that combines explicit scoring dimensions with guided-reasoning prompts, enabling large language models to serve as semantic judges. Its primary contribution is the formalization of semantic evaluation into a repeatable, structured decision process that more accurately reflects human assessment criteria, rather than merely improving scoring consistency [111].

Taken together, multimodal, and alignment-oriented benchmarks extend evaluation beyond task accuracy. They address whether visual-text grounding remains structurally consistent and whether evaluation procedures themselves are institutionally aligned. This shift reinforces the view that enterprise reliability depends not only on correct answers but on structurally grounded reasoning and evaluation mechanisms that remain accountable under cross-modal and regulatory constraints.

6 Governance, Compliance, and Risk Control

Effective governance of RAG and LLM systems in enterprise settings requires more than attribution and faithfulness. It also demands clear responsibility for errors, strong attack detection and mitigation, and the ability to adapt to regulatory changes. This shift reflects a transition from technical trustworthiness to institutional accountability in generative systems. Although governance frameworks and risk control models vary across industries, a consistent requirement is that evaluation and monitoring must be embedded into compliance processes and audit structures to ensure that system behavior can be traced, explained, and controlled.

6.1 Regulatory Frameworks and Governance Models

As generative AI systems are introduced in high-risk domains, enterprises are shifting their focus from model accuracy to the ability to regulate and audit system behavior. This shift marks the transition of generative AI from experimental technology to a field subject to regulatory accountability.

Unlike traditional IT governance, which primarily focuses on data access control and infrastructure security, the risks associated with large language models arise mainly at the semantic and reasoning levels, including hallucination, prompt injection, and reasoning drift. These risks are not merely system vulnerabilities but stem from the inherent uncertainty of model behavior. Consequently, governance of generative AI extends beyond data protection to decision auditability and dynamic risk monitoring. In high-risk settings, technical performance evaluation alone is insufficient to sustain institutional trust; corresponding structures must be established across both engineering and regulatory levels [122].

6.1.1 Institutional Framework Expansion and Engineering Gaps

ISO/IEC 42001:2023, as the first certifiable AI management system standard, extends governance beyond traditional IT controls to full lifecycle AI risk management, requiring organizations to establish mechanisms for risk identification, responsibility allocation, incident reporting, and continuous monitoring. However, the standard operates primarily at the institutional level and does not specify how to validate generative model outputs in real time or how to preserve evidence chains at the engineering level.

Similarly, the EU AI Act subjects high-risk AI systems to mandatory regulatory supervision, requiring the retention of technical documentation, risk assessment records, and traceable decision-making processes. Yet empirical findings from AIReg-Bench indicate that large language models exhibit instability in understanding and complying with regulatory provisions, particularly in reasoning over high-risk obligations and restrictive clauses. This reveals a structural gap: regulatory compliance is not merely a documentation issue, but an engineering problem of semantic reasoning stability. Even when organizations complete formal compliance documentation, institutional declarations may diverge from actual model behavior if the model itself lacks stable compliance reasoning capability [123].

Furthermore, the "Right to Explanation" introduces an additional tension. Regulatory frameworks require transparency and explainability, while enterprises often face constraints related to proprietary protection and model confidentiality. Without engineering-level traceability structures, transparency risks remain at the level of policy statements rather than verifiable technical evidence [124].

6.1.2 Representative Governance Scenarios in Enterprise Deployment

Based on existing regulatory frameworks and empirical findings, four recurring governance scenarios emerge in enterprise deployment of generative AI:

1. Misalignment between institutional declarations and model behavior:

Regulatory compliance requires integrating behavioral monitoring into institutional design. Organizations may declare adherence to ISO 42001 or the EU AI Act, yet without continuous technical validation of actual outputs, compliance remains formal rather than operational [125].

2. Insufficient traceability in high-risk decisions:

Without engineering-level logging of evidence retrieval and versioned tracking mechanisms, transparency cannot be translated into verifiable facts. Although the EU AI Act mandates traceable decision processes, generative model reasoning is typically opaque, and the Right to Explanation faces practical tension between commercial confidentiality and model complexity [126].

3. Mismatch between dynamic model updates and static audit cycles:

Traditional IT audits operate on quarterly or annual cycles, whereas RAG systems and LLM versions may update within weeks. AI governance, therefore, requires a shift from static compliance review toward dynamic oversight mechanisms [122].

4. Difficulty mapping semantic risk into enforceable controls:

Conventional governance frameworks are designed for data security and system stability but lack mechanisms to address semantic generation risk. Findings from AIReg-Bench indicate that regulatory reasoning itself constitutes a technical challenge. Unless semantic risk can be translated into measurable control variables, such as attribution ratios or faithfulness thresholds, regulatory requirements cannot be operationalized at the engineering level [123].

Recent studies increasingly converge on a shared conclusion: the effectiveness of AI governance does not depend solely on regulatory text, but rather on whether regulatory controls can be translated into engineering controls. Risk classification, transparency obligations, and incident reporting procedures must therefore be mapped to implementable technical modules, including data source tagging, decision logging, evidence retrieval records, and version tracking structures. Table 8 illustrates this regulatory-engineering mapping logic by aligning governance requirements with traceable system design elements.

Table 8: Major regulatory frameworks, core control dimensions, and mapping to CMES engineering controls.

Regulatory and Governance Framework	Governance Positioning	Core Control Dimension	Key Regulatory or Standard Requirements	Mapped Engineering Controls
NIST AI RMF (2023)	Voluntary and risk oriented governance framework	Traceability	Identification of model purpose, data sources, and usage scope	Version control; data provenance; evidence logging
		Transparency	Interpretability of model behavior and risk exposure	Reasoning visualization; evidence summaries
		Risk Management	Ongoing AI risk identification and mitigation	Risk tier labeling; output risk scoring
		Incident Response	Internal reporting and remediation mechanisms	Anomaly detection; incident logging; notifications
ISO/IEC 42001 (2023)	Certifiable AI management system standard	Traceability	Lifecycle documentation requirements	End-to-end logging; model/config tracking
		Transparency	Defined roles and decision accountability	Role-based access; responsibility tagging
		Risk Management	Integration with organizational risk management	Risk registers; periodic mitigation review
		Incident Response	Continuous improvement obligations	Incident categorization; root cause analysis
ISO/IEC 23894 (2023)	AI-specific risk management standard	Traceability	Systematic AI risk governance	Decision logs; behavior audit trails
		Risk Management	Reproducible risk assessment	Risk matrices; mitigation mapping
EU AI Act (2024)	Legally binding regulatory framework	Traceability	Technical documentation for high-risk systems	Compliance repositories; usage logs
		Transparency	Right to explanation for affected users	User notification; explainable outputs
		Risk Management	Pre- and post-deployment risk assessment	High-risk mode control; safeguards
		Incident Response	Mandatory serious incident reporting	Regulatory reporting; impact analysis

Governance frameworks and control mappings alone do not constitute substantive compliance capability. When enterprises face regulatory review or third-party audits, the decisive factor is whether traceable evidence can be continuously preserved during system operation and whether each generative decision can be verified and reconstructed. Compliance capability, therefore, shifts from institutional design to audit pipelines and end-to-end traceability mechanisms, requiring governance requirements to be translated into automated logging and verification processes, and transforming policy declarations into computable and auditable engineering facts.

6.2 Compliance Audit Pipelines and Traceability

Current AI governance frameworks mandate traceability, transparency, and incident response. However, unless regulatory principles are concretely mapped to engineering workflows, they remain institutional

declarations rather than enforceable accountability structures. In large language model and RAG deployments, compliance auditing extends beyond recording system operations to reconstructing how a specific output was produced, which data and model versions were involved, and whether the decision process complied with predefined risk controls. Traditional IT logging mechanisms, limited to API calls and timestamps, are insufficient to capture prompt versions, retrieval evidence, model parameters, and risk annotations in high-risk environments.

Automated monitoring and black-box evaluation metrics alone cannot expose strategic evasion, latent backdoors, or embedded bias. Consequently, compliance processes increasingly incorporate Offline Human-in-the-Loop (HITL) mechanisms, enabling auditors to review complete audit bundles, including model and prompt versions, training and retrieval sources, generated outputs, and risk annotations. This layered design balances operational efficiency with institutional accountability in high-risk decision contexts [127].

Beyond accountability, data attribution and error traceability remain critical challenges. When hallucinations or faulty reasoning occur, organizations must trace outputs back to underlying documents or training data sources to enable corrective action. Techniques such as Debias and Denoise Attribution (DDA) improve attribution stability in noisy or biased data environments. Nevertheless, multi-stage fine-tuning and proprietary constraints limit full interpretability and disclosure, indicating that end-to-end traceability must operate alongside version control and the preservation of retrieval evidence to establish legally defensible audit structures [128].

In content dissemination and privacy risk control, compliance auditing introduces additional technical requirements. The Waterfall framework embeds persistent textual watermarks into generated outputs, allowing content provenance and generation responsibility to remain verifiable even after cross-organizational transmission. This "evidence-with-content" design extends compliance proof from internal logs to the document itself, strengthening cross-system auditability [129].

At the model level, advances in instruction fine-tuning and internal representation control enable privacy-preserving and machine unlearning mechanisms that actively suppress sensitive information during generation. These approaches indicate that privacy compliance is no longer limited to data deletion but can be implemented through model-level governance controls [130,131].

Overall, enterprise LLM compliance pipelines increasingly adopt layered architectures combining version tracking, data attribution, evidence preservation, watermarking, and human review checkpoints. While such mechanisms enhance traceability and accountability, they remain primarily post-generation controls. As illustrated in Fig. 12, integrating attribution, privacy safeguards, and automated reporting into a unified audit workflow enables continuous evidence collection and verification across the model lifecycle, forming the operational foundation for subsequent real-time governance agents and guardrail mechanisms:

1. Data Acquisition: source identification and document parsing
2. Preprocessing: version control and knowledge segmentation
3. Security Monitoring: guardrails and PII interception for real-time risk control
4. Knowledge Attribution: claim-source linkage through fact-tracking mechanisms such as DDA
5. Evidence Production: watermarking mechanisms, such as Waterfall, to ensure verifiability and tamper resistance

By integrating data lineage, real-time risk monitoring, attribution analysis, and tamper-resistant evidence generation, this workflow enables enterprises to achieve end-to-end traceability and regulatory compliance across the full lifecycle of LLM deployment. Enabling preventive governance requires embedding real-time governance agents and guardrail mechanisms directly into the model's operational process, shifting risk control to the generation stage. This exposes a structural limitation in current architectures: most

compliance and auditing mechanisms remain external to the model's reasoning workflow rather than embedded within its decision chain. Without the ability to constrain outputs during generation, governance remains reactive rather than preventive.

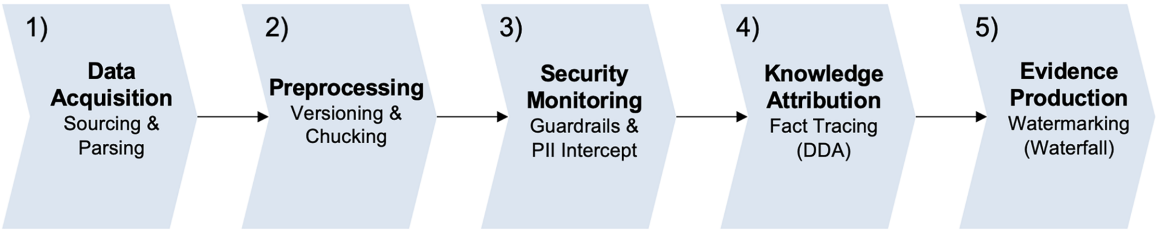

Figure 12: End-to-end traceability and compliance audit workflow for LLM systems.

6.3 Governance Agents and Guardrails for LLMs

While audit pipelines address ex post traceability, this section focuses on ex ante and real-time risk control. In deployment settings, enterprises cannot rely solely on retrospective review, as model errors, jailbreak attempts, prompt injection, or privacy leakage occur at generation time and may immediately trigger legal or commercial risk. Governance, therefore, shifts from recording and reconstruction to real-time prevention and correction.

Fully automated control, however, has inherent limits. Large language models are context-sensitive and probabilistic; static rules cannot cover complex scenarios, and overly restrictive constraints degrade usability. Consequently, Online Human-in-the-Loop (HITL) becomes a design requirement rather than an optional safeguard. As emphasized in the NIST AI RMF, high-impact systems must include escalation pathways that defer uncertain decisions to human oversight.

At the engineering level, governance agents and guardrails constitute the primary real-time defense layer. NeMo Guardrails constrains model behavior through programmable Colang dialogue scripts, limiting outputs to predefined policy boundaries and mitigating prompt-based manipulation [132]. Yet rule-based constraints struggle in semantic gray zones or implicit attack scenarios. A second layer introduces specialized moderation models. WildGuard demonstrates that lightweight task-specific classifiers can more stably detect jailbreak attempts and malicious intent under low-latency conditions [133]. However, such models typically detect known violation patterns and may fail under novel or multi-turn adversarial strategies.

For output consistency, OntoFact aligns generated responses with enterprise KG, enforcing ontology-level constraints as a fact arbitration mechanism [134]. While this strengthens semantic alignment, KG maintenance and update latency limit adaptability in dynamic environments. Accordingly, proactive red teaming has become integral to governance cycles. HarmBench establishes standardized adversarial benchmarks to evaluate robustness under multi-turn attack scenarios, reinforcing the need for continuous stress testing rather than one-time deployment [135,136].

Collectively, real-time governance can be structured into three defense layers: supported by human escalation and adversarial evaluation. These mechanisms must operate as part of an integrated governance architecture rather than isolated tools, extending compliance from documentation and traceability toward continuous, preventive control. Fig. 13 summarizes this end-to-end defense workflow:

1. Pre-generation control: input filtering and safety classification to block prompt injection and policy violations
2. In-generation constraints: semantic boundary enforcement to restrict outputs within predefined domains

3. Post-generation verification: ontology-based fact arbitration, PII masking, and risk-threshold response control

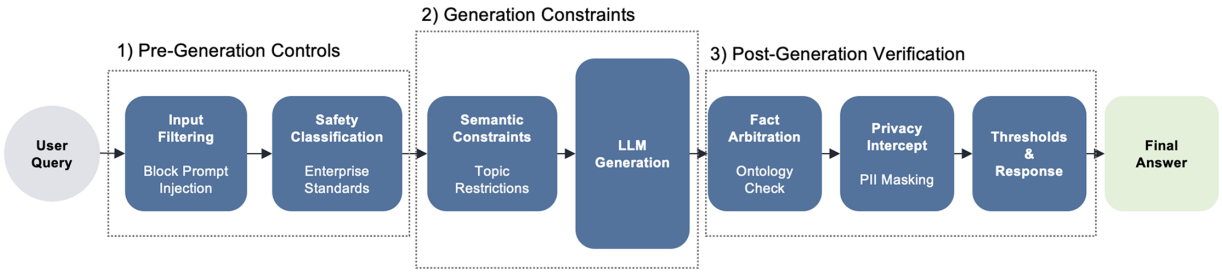

Figure 13: Guardrail workflow for LLM tasks.

7 Key Challenges and Future Directions

When these limitations cannot be fully addressed through architectural refinement alone, the emphasis transitions from isolated performance enhancements to risk control and dynamic self-regulation. Although pipeline-based systems are efficient, they lack the capacity to monitor and revise intermediate results during execution, which increases their vulnerability to error propagation in complex, multi-stage tasks [135,136].

In these contexts, system reliability depends not only on the performance of individual components but also on the continuous validation and adjustment of intermediate outcomes. This necessitates a transition toward execution processes where retrieval, reasoning, and evaluation are coordinated instead of statically arranged. This analysis examines how an agentic workflow, supported by guardrail mechanisms, facilitates coordination by integrating monitoring, verification, and corrective actions into the execution process. This approach operationalizes risk control and dynamic self-regulation within a unified system.

7.1 Complex Document Reasoning & Multi-Modal Grounding

The challenges associated with reasoning over enterprise documents arise not only from model limitations but also from a fundamental disconnect among document structure, semantic distribution, and current RAGs. Unlike open-domain text, enterprise documents often exhibit highly nonlinear organization, cross-page dependencies, multimodal content, and evolving or inconsistent knowledge sources. Recent studies show that visual-language misalignment, layout errors, and knowledge conflicts can propagate through retrieval and generation, leading to unsupported or internally inconsistent outputs [137,138]. Therefore, retrieving relevant content does not necessarily ensure reliable reasoning.

From a system perspective, these issues correspond to core problems identified in Section 2, including the gap between knowledge and reliability, inherent uncertainty in generation, and error propagation across system layers. As indicated in Table 9, these challenges can be categorized into four main areas: cross-page and long-range reasoning, loss of structural semantics, mismatch between reasoning units and citation granularity, and instability of multimodal reasoning chains. The following sections examine the underlying causes of these challenges and their significant implications for system design.

7.1.1 Cross-Page and Long-Range Dependencies: From Local Retrieval to Global Synthesis Failure

The core semantics of enterprise documents are frequently distributed across non-adjacent sections. Elements such as contractual exceptions, financial footnotes, and technical constraints often necessitate

integration across sections or pages to achieve semantic completeness. Prior studies have shown that, in long-document scenarios, critical information may be overlooked due to positional bias and uneven attention allocation, even when the information is within the model's context window [65].

Table 9: Critical research gaps in enterprise document reasoning.

Problem Category	Document Characteristics	Root Cause of Failure	Impact on System
Cross-Page Reasoning (7.1.1)	Fragmented info across pages/sections	Chunking breaks long-range semantics	Missing distant but critical evidence
Loss of Structural Semantics (7.1.2)	Complex spatial layouts (tables/figures)	Serialization flattens document structure	Reasoning deviates from original context
Granularity Mismatch (7.1.3)	Specific field or clause-level citations	Retrieval unit is too coarse (e.g., paragraph)	Imprecise attribution and traceability
Multimodal Instability (7.1.4)	Joint dependency on text and visuals	Poor integration of heterogeneous signals	Biased or incomplete reasoning chains

As shown in Fig. 14, a clear semantic gap between current technical architectures and enterprise requirements. Most RAG systems rely on the chunk independence assumption, segmenting documents into isolated vector units to enhance retrieval efficiency. Although this approach enables approximate search, it disrupts cross-page dependencies and interrupts the continuity of reasoning structures. Consequently, even when relevant evidence is retrieved, it is often not integrated into coherent, traceable reasoning, and retrieval success does not guarantee the reliability of the reasoning [139].

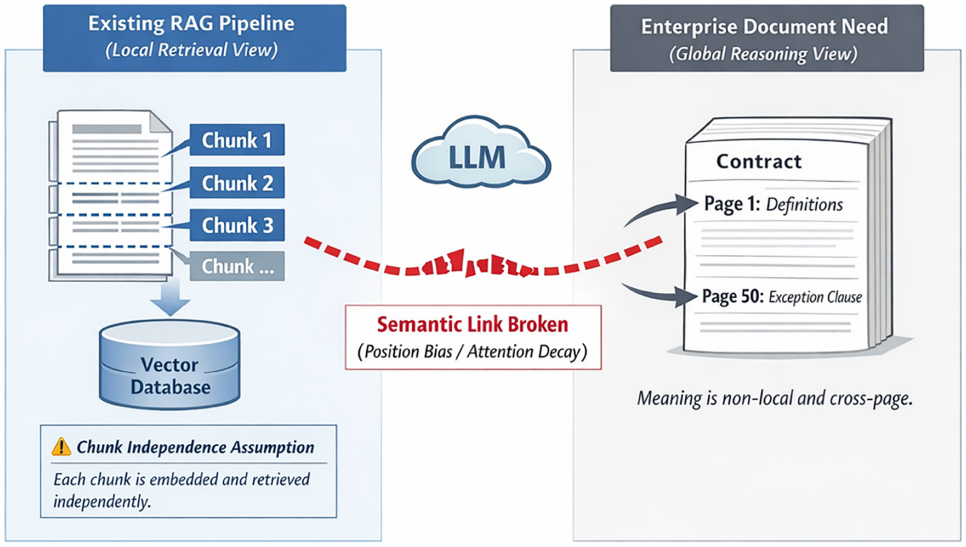

Figure 14: The semantic gap between local retrieval and global reasoning.

At the system-level, such semantic disconnection prevents the integration of dispersed evidence into unified reasoning premises. In enterprise applications, these failures rarely appear as explicit hallucinations;

instead, they manifest as implicit omissions of compliance conditions within otherwise fluent outputs. Existing studies characterize this phenomenon as a structural integration failure rather than a simple generation error [139]. In decision contexts that depend on conditional consistency, such latent fragmentation increases uncertainty in compliance review and risk control.

7.1.2 Structural Semantic Loss: The Challenge of Decoding Layout Language

Tables, hierarchical headings, marginal annotations, and figures in enterprise documents collectively create a multi-dimensional semantic space, where meaning is determined by spatial alignment and structural positioning. In contrast, most current document understanding pipelines depend on linearization, which compresses complex layouts into one-dimensional text sequences and consequently diminishes the structural semantics present in the original document.

Previous research demonstrates that text-only sequence representations fail to adequately preserve semantic signals derived from layout, especially in the context of table alignment, heading hierarchies, and cross-block relationships. Even when layout or visual features are incorporated, models must learn to integrate these with linguistic semantics; otherwise, structural cues are not fully leveraged during reasoning [30]. This indicates that perceiving layout is not equivalent to understanding the language of layout.

A significant limitation arises from the irreversible nature of linearization as an information compression process. When row-column alignment, block boundaries, or hierarchical relationships are diminished during transformation, reconstructing the original structure becomes challenging. Empirical analyses further reveal that, even with the inclusion of structural markers or special tokens, models processing complex layouts frequently default to text-order-dominant reasoning, resulting in superficial utilization of spatial relations [36].

In pipeline-based systems, the loss of structure makes it harder to judge evidence. Even if the right content is found, broken or missing structures can lead to incorrect links between values, conditions, or ranges. In business settings, this often leads to results that sound good but don't match the real logic, making tasks like checking finances, comparing specs, or reviewing contracts less reliable and harder to verify.

7.1.3 Mismatch between Reasoning Units and Citation Granularity: A Structural Conflict between Explainability and Auditability

In enterprise governance and high-risk decision scenarios, correctness is necessary but not sufficient for practical adoption. Compared to open-domain applications, enterprise systems require models to precisely align each key conclusion with specific evidence sources, including contract clauses, regulatory subsections, or individual table cells in financial reports. Therefore, reasoning must be both plausible and traceable, and auditable at a granular level.

However, most existing RAG systems are structured around paragraphs or fixed-length chunks as their primary units for reasoning and generation. This approach introduces a significant gap between the model's reasoning granularity and the fine-grained evidence structures required for enterprise auditing. During generation, models often synthesize information from multiple evidence fragments but do not maintain a one-to-one correspondence with the original evidence units in the output. Consequently, citations typically remain at the paragraph or document level. Prior research demonstrates that even when generated content is semantically correct, models may fail to accurately identify the specific external evidence supporting their conclusions, which undermines explainability and verifiability [97].

This mismatch represents a structural design challenge rather than a singular implementation problem. While larger reasoning units can enhance retrieval stability and semantic coverage, enterprise auditing

requires highly precise localization at the citation level. Recent studies indicate that even when models utilize external evidence during generation, the resulting outputs may not consistently align with specific evidence fragments. As a result, citations often serve as post hoc explanations rather than as integral supports for the reasoning process [140].

In pipeline-based systems, once outputs are generated, any misalignment cannot be corrected. This makes it difficult to verify whether conclusions are based on the correct evidence. Consequently, manual validation becomes necessary, and integration into automated decision workflows is limited. In governance-sensitive environments, where accountability and traceability are critical, this gap restricts system trustworthiness and scalability.

7.1.4 Instability in Multi-Modal Reasoning Chains: Modality Bias and Selective Evidence Forgetting

In enterprise decision-making contexts, textual descriptions, tabular data, and graphical trends collectively constitute the evidentiary foundation for conclusions. While multimodal documents are intended to provide complementary information that reduces uncertainty, systematic evaluations demonstrate that language models often prioritize linguistic fluency and narrative coherence rather than robust integration of cross-modal constraints. As a result, outputs may appear plausible even when they do not fully represent the underlying evidence [141].

This predominance of language reflects a structural asymmetry in modality processing within multimodal reasoning systems. Even when both textual and visual inputs are present, models frequently default to shortcut judgments based on linguistic cues, neglecting constraints imposed by visual evidence. Empirical studies in vision-language understanding reveal that when inconsistencies occur between textual and visual content, models often fail to sustain stable cross-modal alignment, resulting in judgments that correspond more closely to linguistic structure than to actual visual relationships. In these instances, visual evidence is diminished or selectively disregarded within the reasoning process [142].

A key limitation is the lack of mechanisms that explicitly enforce modality balance during inference. When conclusions rely on joint constraints from multiple modalities, models often fail to appropriately weight evidence, frequently relegating structured or visual inputs to secondary roles. Consequently, reasoning outcomes become unstable and sensitive to input presentation and narrative formulation, rather than being consistently grounded in the complete evidence space.

Within system-level workflows, such modality imbalance introduces a form of uncertainty that is difficult to detect and calibrate. Outputs typically remain fluent and internally consistent; modality imbalance is often difficult to detect. Previous socio-technical risk analyses indicate that excessive reliance on surface fluency can obscure underlying grounding limitations, thereby making systematic errors more challenging to identify [143]. As a result, multimodal reasoning in current systems remains difficult to verify and cannot be reliably calibrated for governance- and audit-critical applications.

7.2 System Constraints: Conflict Resolution and Cost Trade-Offs

Enterprise-grade LLM systems reveal system-level constraints involving knowledge conflicts and resource trade-offs, which are closely linked to the reliability and error propagation issues previously discussed. Multi-source documents often exhibit version discrepancies and inconsistent authority hierarchies. In the absence of explicit conflict-handling mechanisms, models may implicitly merge contradictory evidence during generation, yielding fluent but logically inconsistent outputs. Such inconsistencies introduce significant risks in compliance and auditing contexts. To mitigate these risks, systems must implement

authority ranking, version control, and refusal strategies to prevent generated responses from amplifying underlying conflicts.

At the same time, conflict management and iterative retrieval processes increase latency and computational costs, necessitating trade-offs among accuracy, throughput, and hardware resources. Consequently, the challenge for enterprise LLM systems extends beyond model capabilities to include maintaining deployable, governable architectures in the presence of inconsistent information and limited resources.

7.2.1 Conflict Detection and Controlled Resolution

In enterprise RAG systems, knowledge conflict is inherent to multi-source environments. When retrieved evidence derives from different versions, authority levels, or semantically inconsistent sources, the lack of governance mechanisms permits contradictions to enter the generation stage and be rationalized into superficially coherent conclusions. This issue is structural rather than solely model-specific, as inconsistent evidence is incorporated without constraint.

As illustrated in Fig. 15, enterprise conflict handling can be formalized as a structured pipeline in which conflict detection and management occur prior to generation, followed by a controlled output subject to attribution constraints. Retrieved evidence should not be incorporated unconditionally; instead, it must first undergo evaluation for consistency and credibility.

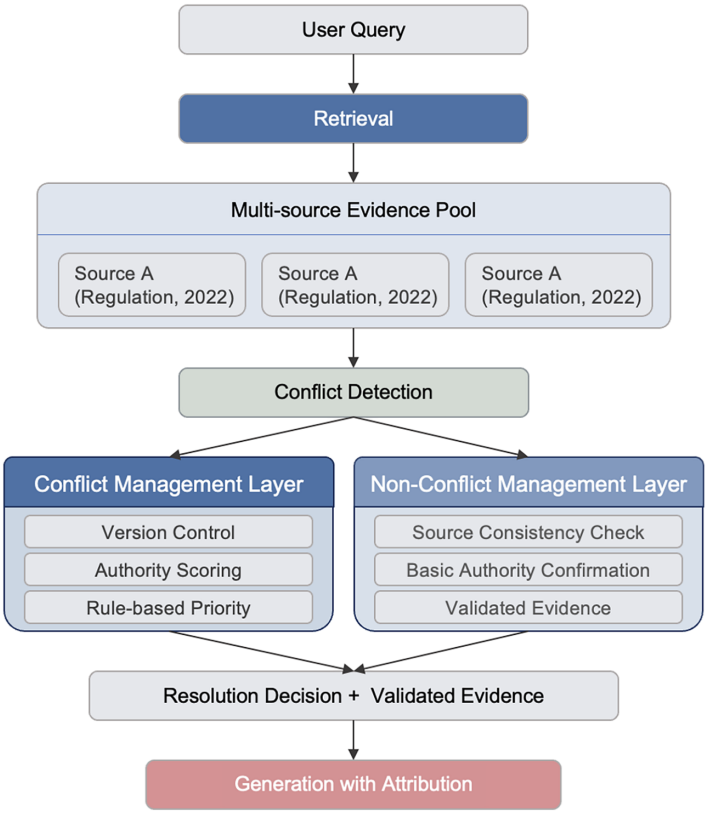

Figure 15: Knowledge-centric multi-agent system architecture for auditable enterprise reasoning.

First, during the conflict detection stage, the central question is how to assess the reliability and sufficiency of evidence. Prior work shows that RAG reasoning errors often stem from fluctuations in retrieval quality and implicit semantic conflicts. Self-RAG introduces a self-reflection mechanism that

prompts the model to critique and reassess evidence before answering, triggering additional retrieval when inconsistencies are detected [109]. CRAG, by contrast, incorporates an independent retrieval-quality evaluator that assigns confidence scores and re-queries when results fall below a threshold [144]. These approaches shift conflict handling forward to the generation or retrieval stage, preventing unreliable evidence from directly influencing reasoning.

When contradictions or version inconsistencies are detected, the system enters a conflict management layer. Rather than forcibly resolving all conflicts, this layer reorganizes, and weights evidence based on authority ranking, temporal alignment, and contextual consistency. In enterprise settings, regulatory updates or cross-department discrepancies may cause temporary inconsistencies; therefore, the system should allow downgrading output or annotating uncertainty rather than enforcing a single definitive answer. This mechanism prevents unstable evidence from being propagated unchecked into the next generation.

In the controlled generation with attribution stage, outputs must adhere to both evidence alignment and consistency constraints. Even after retrieval and management calibration, language models may overgeneralize or improperly merge information due to fluency-driven tendencies. Without post hoc consistency checks and attribution validation, outputs that appear semantically natural may still be based on unstable reasoning foundations [145]. For instance, SelfCheckGPT identifies potential hallucinations by measuring response variance across multiple stochastic generations, while RAGAS assesses faithfulness and context precision to quantify alignment between outputs and retrieved evidence.

In enterprise deployment, the objective of this stage is not merely to improve accuracy but to build auditable output structures. Concretely, conclusions must map to identifiable evidence fragments; conflicting sources require uncertainty or version annotation; and reasoning must remain within the calibrated evidence boundary. By embedding attribution and consistency constraints into the generation process, systems transform linguistic plausibility into evidence-controlled reasoning, thereby maintaining auditability and decision-level credibility under multi-source inconsistency and dynamic updates. Such conflicts, if unresolved, can propagate through subsequent reasoning and undermine both the consistency and reliability of system outputs.

7.2.2 Cost-Latency-Reliability Trade-Off

In enterprise-grade LLM systems, performance bottlenecks primarily result from engineering interdependencies among retrieval depth, long-context reasoning, and deployment strategies, rather than from isolated model deficiencies. Previous studies demonstrate that scaling to billion-scale vector indices and extended context windows leads to nonlinear increases in latency and resource consumption. Performance degradation is frequently attributed to module chaining and data movement overhead, rather than to single-model inference time alone [146]. Recent research indicates that this engineering tension can be systematically analyzed across three layers: retrieval, reasoning, and deployment, as summarized in Table 10.

Table 10: Engineering tradeoffs among cost, latency, and reliability in enterprise LLM systems.

Engineering Layer	Technical Adjustments	Reliability Impact	Latency & Cost	Core Risks
Retrieval	top-k rises, multi-hop, larger vector index	Better coverage & consistency	I/O & memory overhead increases	Retrieval noise & error accumulation

(Continued)

Table 10 (continued)

Engineering Layer	Technical Adjustments	Reliability Impact	Latency & Cost	Core Risks
Context & Reasoning	Extended context length; optimized attention	Enhanced long-range dependency	Quadratic cost & memory pressure	Positional bias
Model Deployment	Quantization (QLoRA/GPTQ), caching	Potential numerical instability	Reducing hardware &inference costs	Accuracy fluctuation; fragile reasoning

At the retrieval layer, increasing the top-k parameter or implementing multi-hop retrieval enhances evidence coverage and cross-document consistency, but also increases the number of vector comparisons and memory access costs. Although GPU acceleration can significantly improve approximate nearest neighbor search, input/output and memory management remain primary bottlenecks at billion-scale indexing. Consequently, deeper retrieval improves reliability but incurs higher latency and hardware costs. In the absence of adaptive retrieval control, expanding evidence pools may introduce semantic noise and diminish ranking stability.

At the reasoning layer, attention computation scales quadratically with sequence length, making extended context windows computationally expensive. FlashAttention [147] reduces memory access overhead by optimizing attention computation, thereby improving the feasibility of long sequences. Furthermore, efficient Transformer architectures, such as the Long-Short Transformer, integrate local and global attention patterns to reduce the computational demands of long-context processing while maintaining the capacity for long-range dependency modeling [148]. Nevertheless, expanding the context window continues to introduce additional latency and memory pressure in real-time systems. Consequently, context window expansion should be aligned with task risk levels and service-level agreements (SLA) rather than applied indiscriminately.

At the deployment layer, quantization and compression techniques are essential for cost control. QLoRA [149] integrates 4-bit NF4 quantization with low-rank adaptation to reduce memory usage and fine-tuning costs, while GPTQ [150] employs Hessian-based post-training quantization to maintain inference quality under constrained GPU resources. However, low-bit inference can amplify numerical errors in extended reasoning chains or multi-step evidence integration, thereby degrading the precision of high-precision tasks. As a result, quantization functions not only serve as a cost-optimization mechanism but also affect system reliability.

Furthermore, overall latency in multi-stage RAG pipelines often stems from coordination among the retrieval, reranking, and generation components, rather than from the performance of individual modules. Lewis et al. [16] observe that in the absence of coordination mechanisms, evidence stitching, and context reconstruction introduce hidden overhead. This cross-module accumulation results in nonlinear performance degradation as system scale increases. Consequently, enterprise deployment relies more on controlled resource orchestration and stable module interaction than on optimizing individual modules.

Cost, latency, and reliability do not function as independently optimizable objectives; rather, they are interdependent variables that must be managed jointly. Enterprise deployment, therefore, requires deliberate trade-offs among retrieval depth, context length, and model precision under task-specific risk and resource constraints. The effectiveness of this coordination determines whether a system can move beyond

experimental settings toward stable operational use, as these trade-offs influence not only system efficiency but also the stability and reliability of reasoning outcomes under real-world constraints.

7.3 Research Blueprint for Reliable and Governable Systems

The natural scaling of model size does not drive the evolution of enterprise-grade LLM systems; rather, it is a series of structural tensions encountered during real-world deployment. These tensions arise from challenges in ensuring that retrieved knowledge supports reliable, verifiable reasoning, as well as from the tendency for errors to propagate across interconnected system components. As language models transition from experimental settings to enterprise decision-making workflows, the primary concern shifts from generating correct answers to maintaining stable, accountable, and traceable system behavior amid dynamic data updates, diverse evidence integration, and regulatory requirements. This shift from capability to responsibility is the principal catalyst for advancing enterprise LLM systems.

Fig. 16 illustrates this transformation as a three-stage progression in architectural focus: Model-Centric Optimization, Pipeline-Level Coordination, and ultimately Structure-Centric Governance. The figure highlights not only shift in technical priorities but also the progressive formalization of risk propagation paths and control boundaries across architectural layers. As systems evolve, risks are no longer confined to isolated modules but propagate along the end-to-end pipeline, requiring explicit monitoring and optimization signals at each layer. At each stage, knowledge gaps, module coupling, and compliance-driven traceability pressures represent the predominant risks. Simultaneously, challenges shift from single-output quality to consistency control and lifecycle auditing. This progression reflects a transition from focusing solely on enhancing individual model capabilities to systematically managing the occurrence, propagation, and containment of errors within enterprise LLM systems. The maturity of these systems is increasingly defined by the incorporation of governance mechanisms into their architecture, rather than by model size alone.

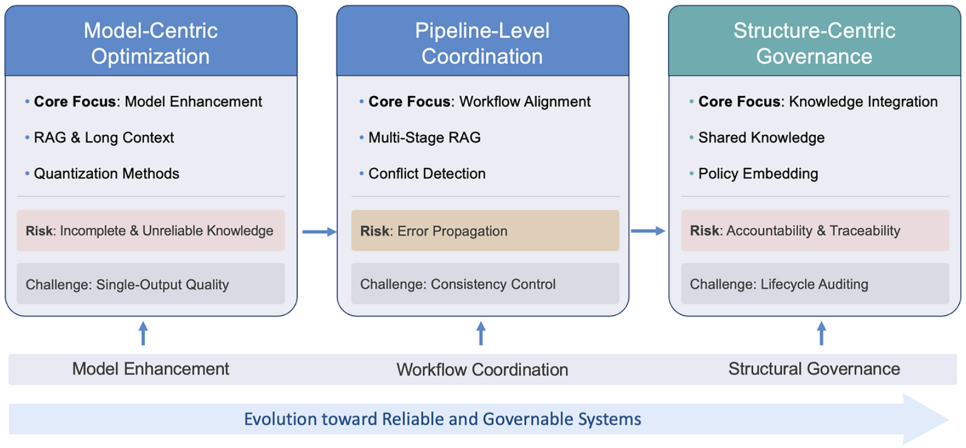

Figure 16: Evolutionary trajectory of enterprise LLM systems.

7.3.1 Model-Centric Optimization

During the initial phase of enterprise deployment, the primary challenge is achieving comprehensive knowledge coverage and minimizing update latency. Pretrained models depend on static parametric memory, while enterprise knowledge bases are subject to continuous evolution. Consequently, model outputs may exhibit linguistic fluency but fail to incorporate updated contractual clauses, cross-document dependencies, or new regulatory constraints. Early research addressed this knowledge gap by focusing on extending model capabilities.

RETRO [151] incorporates large-scale external retrieval into the token generation process. By retrieving relevant passages from an external corpus for each token and conditioning generation on these contexts, RETRO reduces reliance on parametric memory and improves factual accuracy. FlashAttention-2 [152] addresses computational bottlenecks in long-context attention by optimizing parallelization, which enables practical extended context reasoning. Additional research explores long-context fine-tuning, retrieval strategy optimization, specialized reasoning in smaller models, and alternative sequence architectures [153–156].

Despite these advances, improvements at this stage are primarily limited to the quality of individual outputs. Although models demonstrate increased knowledge and context awareness, they lack robust mechanisms for reconciling subtle inconsistencies across multiple sources. Addressing knowledge gaps alone does not resolve structural integration challenges. This stage highlights that increasing model capacity alone does not resolve the reliability of retrieved or inferred knowledge.

7.3.2 Pipeline-Level Coordination

As RAG and multi-step reasoning pipelines become more modular, module coupling introduces new sources of instability. Dependencies among retrieval, reranking, generation, and verification modules increase in complexity, allowing errors introduced at earlier stages to propagate across the pipeline and affect downstream reasoning outcomes. Therefore, system reliability becomes dependent on workflow alignment rather than solely on individual model performance.

Self-Refine [155] implements an iterative generate-evaluate-revise loop, allowing models to reflect on and improve their outputs. DSPy [156] conceptualizes language model invocations as declarative components, which can be compiled into structured pipelines, thereby transforming multi-step reasoning into workflows that are both analyzable and optimizable. Further research enhances pipeline-level control through automated retrieval evaluation frameworks and tool-learning mechanisms [157–159].

As system complexity increases, new risks emerge, such as hidden dependencies, latency accumulation, and cascading inconsistencies. Local optimization of individual modules does not guarantee global consistency. The central challenge at this stage is consistency control, which involves ensuring that decisions remain internally coherent across multi-stage workflows.

7.3.3 Structure-Centric Governance

When enterprise LLM systems are deployed in high-stakes domains such as contract interpretation, financial auditing, or regulatory compliance, the primary concern shifts from workflow consistency to institutional accountability and traceability. At this stage, the main risk is not model error itself, but the inability to ensure traceability, reproducibility, and clear responsibility allocation throughout the system lifecycle.

SWE-agent [160] integrates language models into persistent task environments, maintaining explicit state tracking and action histories to support long-term traceability. The NIST Generative AI Profile [161] further extends AI risk management principles to generative AI systems by emphasizing risk identification, measurement, mitigation, and governance across the system lifecycle. Related research and management standards underscore the importance of shared knowledge structures and audit-ready decision traces as foundational requirements for enterprise AI systems [162,163].

At this stage, system maturity is determined not by incremental improvements in generative performance, but by the capacity to maintain stable, traceable, and policy-aligned behavior across changing data, models, and regulatory environments. The central challenge becomes lifecycle-level auditing, which ensures that decisions remain reproducible and institutionally accountable over time, highlighting a structural limitation of purely pipeline-based coordination. While workflow alignment improves local consistency, it

does not inherently provide a stable semantic foundation for cross-document reasoning and long-term policy control. As enterprise deployments scale, more explicit structural representations become necessary. The next section, therefore, examines how KGs re-emerge within enterprise LLM architectures as mechanisms for semantic consistency and governance support.

7.4 Revisiting Knowledge Graphs in Enterprise LLM System

As challenges related to document-level reasoning, citation granularity, and multimodal alignment become increasingly apparent, the core bottleneck of enterprise-grade LLM systems shifts from model capability to the architecture's ability to ensure reliable, verifiable, and consistent reasoning under dynamic conditions. The central concern is no longer solely performance, but whether the system can effectively identify and control how risks propagate across retrieval and generation processes and enforce clear control boundaries. Within this context, KGs re-emerge as structured solutions deserving careful reassessment. Compared with document-oriented RAG architectures, KGs explicitly represent entities, relationships, and constraints, thereby providing a transparent semantic framework for reasoning. Nevertheless, the significant costs associated with constructing and maintaining KGs limit their feasibility as a universal foundation for all enterprise systems.

The central issue is not whether KGs are inherently superior to RAG, but rather under which circumstances dependence on document retrieval and generative flexibility introduces systemic risk. As systems transition from information querying to decision support, the demands for consistency, verifiable reasoning pathways, version governance, and verification and repair mechanisms correspondingly intensify. In this context, structured knowledge provides a means to constrain reasoning, reduce error propagation, and support verifiable decision-making in enterprise environments.

7.4.1 Knowledge Graphs as a Governance Backbone

In enterprise-grade LLM systems, the principal risk stems not from isolated reasoning errors but from the absence of mechanisms to identify which response represents the system's formal position. In document-centric retrieval and generation architectures, outputs are highly dependent on the retrieved evidence and the contextual configuration during inference. Even when responding to identical business queries, differences in retrieval scope or generation pathways can produce inconsistent interpretations. In compliance-sensitive, contractual, or policy-governed environments, such variability poses uncertainty in accountability and complicates responsibility allocation.

In this context, KGs serve as a foundational element for governance. By explicitly encoding entities, relationships, and constraint structures within a structured representation, the system can distinguish descriptive knowledge from normative knowledge that requires enforcement. This distinction establishes stable semantic boundaries for reasoning and reduces dependence on linguistic plausibility alone. In contrast to purely generative alignment, graph-based representations offer a persistent structural reference, enabling outputs to be evaluated against consistent constraints and supporting both auditing and institutional traceability.

Recent research has integrated knowledge graphs into the reasoning core of large language models to improve governability and verifiability. The Think-on-Graph approach [164] constrains inference to graph-structured entity-relation paths, converting reasoning trajectories into artifacts that can be inspected. Reasoning on Graphs [165] builds on this by emphasizing alignment between intermediate reasoning steps and the underlying graph topology, thereby reducing hidden assumptions and hallucinated inferences. Collectively, these studies demonstrate that, for enterprise systems, plausibility is insufficient; reasoning must be structurally grounded and verifiable to constitute the system's formal position.

From a governance perspective, embedding knowledge graphs as the reasoning substrate transforms the construction of institutional positions within enterprise LLM architectures. Knowledge graphs are no longer limited to retrieval augmentation or performance optimization. Instead, they establish structural foundations for preserving consistency, allocating responsibility, and ensuring normative compliance. In high-risk, long-term operational contexts, this structural anchoring is essential for maintaining controllability and governance stability.

7.4.2 Hybrid RAG-KG Architectures

In enterprise contexts, not all knowledge lends itself to complete structural formalization. Highly context-dependent and frequently evolving document content is difficult to fully incorporate into a knowledge graph without incurring significant construction and maintenance costs. As a result, most systems continue to rely on document-oriented RAG architectures as their foundational framework. However, when reasoning spans multiple documents or maintains fact-level coherence, document-sliced approaches may fail to capture essential relational signals, thereby undermining the consistency of the generated outputs.

To address this challenge, recent research has explored hybrid RAG-KG architectures that incorporate knowledge graphs as a complementary structural layer, rather than as a complete substitute for document-oriented workflows. As summarized in Table 11, document-based RAG and structured knowledge graphs demonstrate strong complementarity regarding knowledge properties and reasoning logic. Enterprise systems can achieve both flexibility and rigor by coordinating these approaches: RAG manages highly dynamic information, while knowledge graphs explicitly represent key factual relationships to constrain reasoning, improve cross-document consistency, and support more reliable and verifiable outputs.

Table 11: Comparative characteristics and hybrid synergies of RAG and knowledge graphs in enterprise systems.

Feature	Document-Centric RAG	Knowledge Graph (KG)	Hybrid Synergies
Knowledge Characteristics	Dynamic, unstructured, and descriptive narrative content	Stable, rule-oriented entities and authoritative relations	Combines structured stability with dynamic contextual details
Reasoning Characteristics	Probabilistic and flexible for generation; prone to discontinuity	Deterministic and logic-driven with verifiable reasoning paths	KG guides retrieval paths to reduce hallucinations and inconsistency
Maintenance Cost	Low; leverages raw documents without extensive upfront modeling	Higher; requires ontology design and continuous structured updates	Supports incremental evolution from RAG toward KG extraction
Granularity of Attribution	Coarser; typically, at the paragraph or chunk level	Fine-grained; enables precise entity or triple-level attribution	Facilitates precise evidence tracing via structural alignment

KG2RAG [166] exemplifies this approach by leveraging knowledge graphs to explicitly encode entities and factual relationships. This guides the organization and expansion of retrieved evidence, thereby enhancing cross-document coherence during generation. These methods indicate that the primary value

of structured knowledge lies not in exhaustively modeling all information, but in the minimal explicit representation of essential relationships.

From the perspective of enterprise systems, hybrid RAG-KG architectures represent a pragmatic compromise. Descriptive and rapidly changing information continues to be managed through document-oriented retrieval, while stable relationships that significantly affect reasoning outcomes are gradually introduced in structured form. This incremental approach enables knowledge graph integration to evolve in alignment with system requirements, avoiding the risks associated with large-scale implementation at initial deployment.

7.4.3 From Static Graphs to LLM-Augmented Knowledge Structures

A significant challenge for knowledge graphs in enterprise systems lies not in their ability to enhance reasoning quality, but in the structural misalignment between their adoption models and the rapidly changing nature of enterprise operations. Conventional knowledge graph deployment requires comprehensive entity definition, ontology design, and consistency validation prior to implementation. In environments characterized by evolving requirements and continuously changing data, this initial engineering effort frequently delays deployment and has been identified as a primary obstacle to enterprise adoption [167].

The emergence of large language models fundamentally changes this limitation by allowing knowledge structures to be incrementally constructed during system operation. In practice, large language models facilitate incremental ontology refinement and staged relation extraction by integrating newly observed document evidence into the knowledge graph. For instance, in contract analysis, newly introduced clauses or regulatory updates can be incrementally formalized as structured relations, eliminating the need for complete schema redesign. However, this flexibility introduces new challenges. Conflicts may occur between relatively stable knowledge graph representations and continuously updated document corpora, and unsophisticated updates may propagate inconsistencies throughout downstream reasoning processes.

To address these challenges, systems implement staged extraction and arbitration mechanisms that assess candidate relations based on recency and source authority prior to integration. This demand-driven approach reduces construction and maintenance costs relative to comprehensive upfront modeling but shifts complexity toward ongoing validation and governance. Wang et al. [168] exemplify this approach with Editable Memory Graphs, which allow knowledge to be dynamically inserted, updated, and constrained during generation. More broadly, structured representations facilitate the externalization of intermediate reasoning states, thereby enhancing transparency and reducing hidden assumptions and uncontrolled error propagation [169,170].

Consequently, knowledge graphs are not supplanted by large language models but are instead redefined as adaptive structural components that balance flexibility with controlled knowledge evolution. Although this paradigm enhances scalability and alignment with dynamic enterprise data, its effectiveness ultimately relies on the system's capacity to enforce consistent update policies and prevent the accumulation of structural inconsistencies over time.

7.4.4 Multi-Agent Collaboration and Knowledge-Centric Coordination

As enterprise-grade LLM systems evolve from single-reasoning pipelines to multi-agent collaborative architectures, the primary challenge shifts from individual model capability to the ability of multiple agents to sustain coherent and controllable collective behavior over extended periods. Role specialization, including retrieval, reasoning, verification, and monitoring, enhances modularity and task coverage within multi-agent

architectures. However, these benefits come with structural risks, especially when shared states and semantic baselines are not explicitly defined.

A significant concern is inconsistency in shared cognitive states. When agents primarily depend on their own contextual windows and locally generated outputs, their interpretations of identical entities or rules may diverge over time. In cross-step reasoning or long-term tasks, this divergence directly weakens system-level consistency and creates instability that is hard to detect and fix once it spreads across agents, and the architecture might lack clear criteria for defining the system's formal interpretation.

Multi-agent systems encounter challenges in achieving consensus. When agents base their reasoning on different document versions or retrieval outcomes, coordination that relies exclusively on linguistic exchange can increase ambiguity, permitting local assumptions to be misinterpreted as global conclusions. Additionally, reasoning biases introduced by one agent may be recursively adopted by others, leading to error propagation across agents and reinforcing flawed reasoning loops, which are often only detected after significant output deviations.

Within this context, knowledge-centric coordination offers a structural reference point. By explicitly representing core entities, relationships, and states within a structured knowledge layer, such as a knowledge graph, agents are provided with stable semantic anchors that mitigate cognitive drift and state divergence. This design principle is consistent with research that constrains LLM reasoning within knowledge graph structures [164,165]. When reasoning is limited to structured relations, the knowledge layer acts as a behavioral boundary, ensuring traceable reasoning paths and preventing the spread of intermediate errors across agents.

As shown in Fig. 17, the architecture establishes the KG as a governance backbone, providing a shared cognitive layer for retrieval, reasoning, and verification agents. Through KG-guided control mechanisms, dynamic document information is integrated with stable logical constraints, facilitating decision-support outputs that maintain traceable reasoning paths and detailed attribution.

While this formulation clarifies the role of knowledge graphs as a coordination backbone, it remains conceptual and does not fully define agent interaction or the operationalization of coordination. To address these gaps and better mitigate cognitive drift and error propagation, Fig. 18 introduces knowledge-centric coordination through three mechanisms: semantic alignment, controlled knowledge evolution, and auditable reasoning.

1. Entity-Anchored Interaction Protocol: To reduce cognitive drift, agents communicate using shared knowledge graph identifiers instead of unstructured language. Operations like Query_KG (EntityID) and Validate_Reasoning (Path) allow agents to reference consistent entity representations. This ensures a unified semantic foundation and supports structured multi-agent coordination [171,172].
2. Staging-to-Arbitration Update Workflow: To balance dynamic updates with knowledge graph stability, extracted entities and relations are first sent to a staging buffer. A conflict arbitration module then evaluates these elements before integration. This staged process supports incremental knowledge evolution and manages conflicts between recency and source authority. The workflow aligns with LLM-assisted knowledge graph construction approaches [173].
3. Hierarchical Traceability and Audit Mechanism: To address error propagation, each reasoning step is documented as a structured subgraph trace. A verification agent evaluates intermediate outputs against knowledge graph constraints. This mechanism enables backward tracing from final outputs to specific reasoning steps and knowledge artifacts. It also facilitates error tracing across agents and clear assignment of responsibility, thereby improving the reliability of multi-step reasoning.

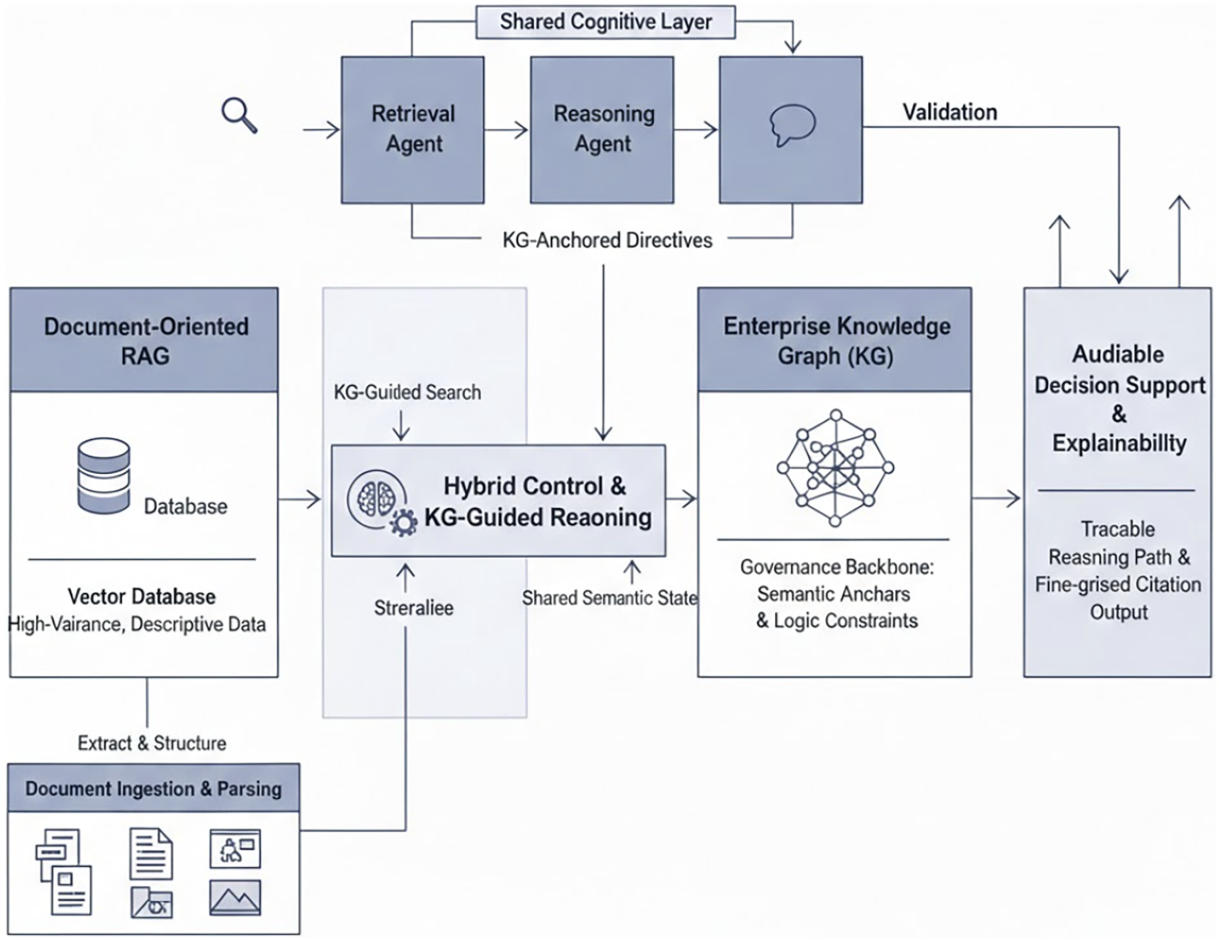

Figure 17: Knowledge-centric multi-agent system architecture for auditable enterprise reasoning.

The main challenge in multi-agent systems is to maintain consistency, controllability, and traceability while allowing flexibility, rather than focusing solely on autonomy. In high-risk, long-term operations, structured knowledge supports coordinated behavior and helps manage system-level risk through semantic alignment, controlled knowledge evolution, and auditable reasoning as described above.

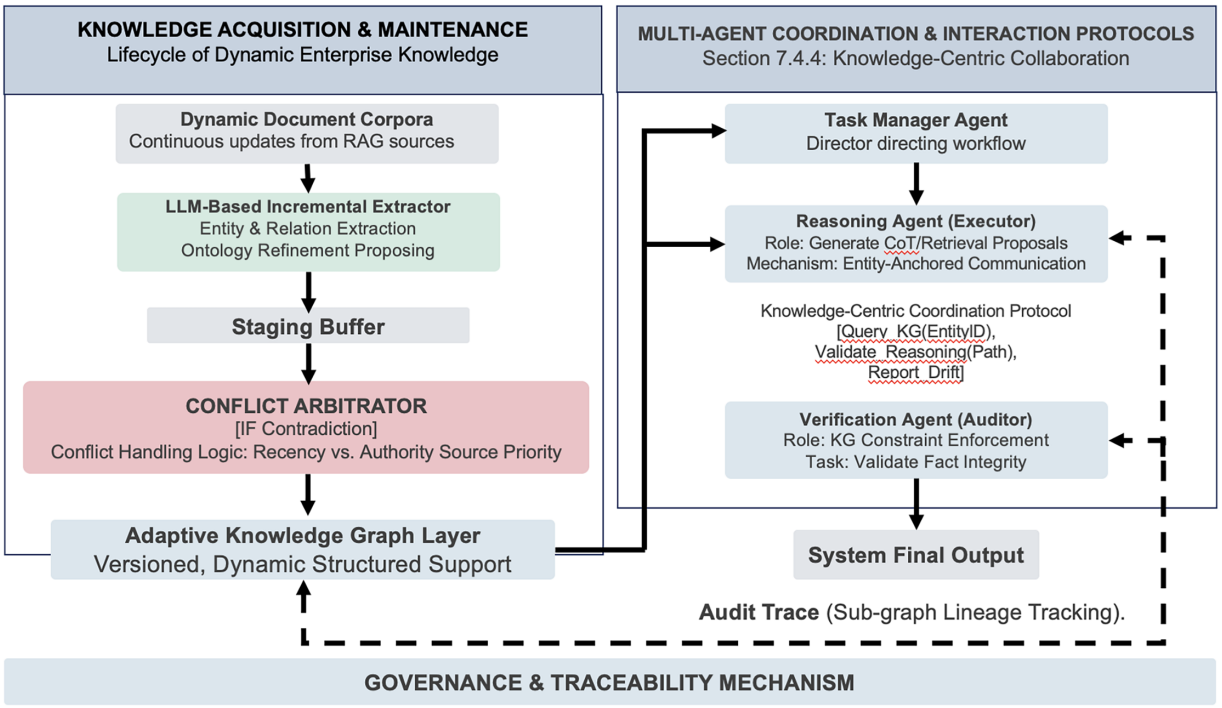

Figure 18: Implementation blueprint of multi-agent coordination.

8 Conclusion

This study is motivated by the practical requirements of enterprise-grade LLM systems and systematically reviews the technological evolution from document perception and RAG-based reasoning to evaluation safeguards and multi-agent workflows. We contend that the primary bottleneck in enterprise adoption has shifted from model selection to the structural disconnect between fragmented system components and the requirement for high-reliability, traceable decision-making.

The success of current enterprise applications is attributable not to the intelligence of the models alone, but to the integration of perception, reasoning, and governance layers that establish stable and traceable connections. In the absence of explicit attribution mechanisms or robust safeguards, even high single-task performance is insufficient to warrant integration into core decision-making processes. A trustworthy enterprise system should therefore be conceptualized as an end-to-end trust stack, rather than a collection of isolated technical modules. These challenges highlight the need for a structured framework that directly links system-level bottlenecks to architectural solutions. The trust stack supports risk-managed, continuously improving system behavior within operational constraints.

8.1 Summary and Technical Implications

This study proposes a deployment methodology for enterprise-grade LLM systems centered on risk control and continuous improvement. Rather than treating LLM adoption as a model-centric problem, we redefine it as an end-to-end structural engineering challenge: establishing traceable reasoning chains across fragmented modules and maintaining decision consistency under dynamic data conditions.

The proposed framework consists of four interconnected layers, each targeting a specific system-level bottleneck. The data perception layer reduces document heterogeneity by using document understanding and structural processing to convert complex enterprise documents into consistent evidence units. The

reasoning structure layer, based on a hybrid RAG-KG architecture, ensures that retrieval success leads to reliable reasoning and promotes semantic coherence across documents. The evaluation layer improves attribution and observability by applying faithfulness-based, fine-grained attribution mechanisms, turning hallucination and reasoning biases into explicit risk signals. The governance layer limits error propagation and system inconsistency by integrating guardrails and verification-repair mechanisms within multi-agent workflows. This allows risk signals to guide retrieval, indexing, and prompt configurations, creating a closed-loop adjustment process.

In essence, the core contribution of this study is not the introduction of isolated technical modules, but the construction of a risk-controlled data flywheel architecture. Each system operation produces both reasoning outputs and diagnostic signals; governance mechanisms transform those signals into structural refinements that enhance stability and consistency in subsequent iterations.

Through this process, risk is not only mitigated but continuously monitored and reintegrated into system optimization, enabling sustained adaptation under evolving data and regulatory conditions. The ultimate objective of enterprise LLM systems is not maximal generative capability, but rather traceable, accountable, and continuously improving decision support under high-risk, long-term operational conditions.

8.2 Future Research Outlook

While this study presents a risk-controlled and self-improving architectural framework for enterprise-grade LLM systems, significant challenges remain in sustaining long-term stability, adaptability, and governability in real-world environments. As document streams change, regulatory requirements evolve, and multi-agent interactions become more complex, preserving structural coherence and attribution fidelity becomes more difficult. Mechanisms for detecting structural drift, updating knowledge representations, and preventing the propagation of localized reasoning errors across workflows require further development to ensure sustained deployment.

Enterprise environments consistently face constraints in data security, access control, and confidentiality. These systems handle sensitive contractual, financial, and regulatory information. As a result, feedback and optimization mechanisms must uphold data protection. This need creates tension between adaptability and strict governance.

Future research should focus on feedback-driven architectures that are both adaptive and controllable. These systems must use risk signals effectively while avoiding the creation of new vulnerabilities. It is also essential to test these architectures in multilingual document environments, especially for cross-national enterprises. Differences in language, format, and regulatory context may affect stability and attribution fidelity. The primary goal is to maintain enterprise LLM systems that are verifiable, accountable, secure, and structurally resilient over time. This is more important than simply increasing generative autonomy.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the National Science and Technology Council (NSTC) of Taiwan, under grants NSTC 114-2221-E-025-003-MY3.

Author Contributions: Conceptualization, Yenjou Wang and Jia-Wei Chang; methodology, Yenjou Wang and Chihtan Cheng; investigation, Yenjou Wang and Chihtan Cheng; writing—original draft preparation, Yenjou Wang and Chihtan Cheng; writing—review and editing, Yenjou Wang, Chihtan Cheng and Jia-Wei Chang; supervision, Jia-Wei Chang. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable. This review paper, as no new datasets were generated or analyzed during the current study.

Ethics Approval: Not applicable.

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

Abbreviations

LLM	Large Language Model
RAG	Retrieval-Augmented Generation
VLM	Visual-Language Model
KG	Knowledge Graph
OCR	Optical Character Recognition
AIS	Attributable Information System

References

1. National Institute of Standards and Technology. Artificial intelligence risk management framework (AI RMF 1.0). Gaithersburg, MD, USA: United States Department of Commerce; 2023.
2. Weidinger L, Uesato J, Rauh M, Griffin C, Huang PS, Mellor J, et al. Taxonomy of risks posed by language models. In: Proceedings of the 2022 ACM Conference on Fairness Accountability and Transparency; 2022 Jun 21–24; Seoul, Republic of Korea. p. 214–29. doi:10.1145/3531146.3533088.
3. Solaiman I, Talat Z, Agnew W, Ahmad L, Baker D, Blodgett SL, et al. Evaluating the social impact of generative AI systems in systems and society. arXiv:2306.05949. 2023. doi:10.48550/arxiv.2306.05949.
4. Kadavath S, Conerly T, Askell A, Henighan T, Drain D, Perez E, et al. Language models (mostly) know what they know. arXiv:2207.05221. 2022. doi:10.48550/arxiv.2207.05221.
5. Geirhos R, Jacobsen JH, Michaelis C, Zemel R, Brendel W, Bethge M, et al. Shortcut learning in deep neural networks. Nat Mach Intell. 2020;2(11):665–73. doi:10.1038/s42256-020-00257-z.
6. Dziri N, Kamalloo E, Mathewson K, Zaiane OR. Faithfulness in natural language generation: a survey. Comput Linguist. 2022;48:791–839.
7. Raji ID, Smart A, White RN, Mitchell M, Gebru T, Hutchinson B, et al. Closing the AI accountability gap: defining an end-to-end framework for internal algorithmic auditing. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency; 2020 Jan 27–30; Barcelona, Spain. p. 33–44. doi:10.1145/3351095.3372873.
8. Mitchell M, Wu S, Zaldivar A, Barnes P, Vasserman L, Hutchinson B, et al. Model cards for model reporting. In: Proceedings of the Conference on Fairness, Accountability, and Transparency; 2019 Jan 29–31; Atlanta, GA, USA. p. 220–9. doi:10.1145/3287560.3287596.
9. Breck E, Cai S, Nielsen E, Salib M, Sculley D. The ML test score: a rubric for ML production readiness and technical debt reduction. In: Proceedings of the 2017 IEEE International Conference on Big Data; 2017 Dec 11–14; Boston, MA, USA. p. 1123–32. doi:10.1109/BigData.2017.8258038.
10. Passi S, Jackson SJ. Trust in data science: collaboration, translation, and accountability in corporate data science projects. Proc ACM Hum Comput Interact. 2018;2:1–28. doi:10.1145/3274405.
11. Sloane M, Moss E, Awomolo O, Forlano L. Participation is not a design fix for machine learning. In: Proceedings of the Equity and Access in Algorithms, Mechanisms, and Optimization; 2022 Oct 6–9; Arlington, VA, USA. p. 1–6. doi:10.1145/3551624.3555285.
12. Mökander J, Axente M, Casolari F, Floridi L. Conformity assessments and post-market monitoring: a guide to the role of auditing in the proposed European AI regulation. Minds Mach. 2022;32(2):241–68. doi:10.1007/s11023-021-09577-4.
13. Amershi S, Weld D, Vorvoreanu M, Fourney A, Nushi B, Collisson P, et al. Guidelines for human-AI interaction. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems; 2019 May 4–9; Glasgow, Scotland, UK. p. 1–13. doi:10.1145/3290605.3300233.
14. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language models are few-shot learners. Adv Neural Inf Process Syst. 2020;33:1877–901.

15. Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, et al. PaLM: scaling language modeling with pathways. J Mach Learn Res. 2023;24(240):1–13.
16. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. Adv Neural Inf Process Syst. 2020;33:9459–74.
17. Ram O, Levine Y, Dalmedigos I, Muhlgay D, Shashua A, Leyton-Brown K, et al. In-context retrieval-augmented language models. Trans Assoc Comput Linguist. 2023;11(3):1316–31. doi:10.1162/tacl_a_00605.
18. Shi W, Min S, Yasunaga M, Seo M, James R, Lewis M, et al. REPLUG: retrieval-augmented black-box language models. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2024 Jun 16–21; Mexico City, Mexico. p. 8371–84. doi:10.18653/v1/2024.naacl-long.463.
19. Lin S, Hilton J, Evans O. TruthfulQA: measuring how models mimic human falsehoods. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics; 2022 May 22–27; Dublin, Ireland. p. 3214–52. doi:10.18653/v1/2022.acl-long.229.
20. Ji Z, Lee N, Frieske R, Yu T, Su D, Xu Y, et al. Survey of hallucination in natural language generation. ACM Comput Surv. 2023;55(12):1–38. doi:10.1145/3571730.
21. Dziri N, Lu X, Sclar M, Li XL, Jiang L, Lin BY, et al. Faith and fate: limits of transformers on compositionality. Adv Neural Inf Process Syst. 2023;36:70293–332. doi:10.52202/075280-3081.
22. Mallen A, Asai A, Zhong V, Das R, Khashabi D, Hajishirzi H. When not to trust language models: investigating effectiveness of parametric and non-parametric memories. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics; 2023 Jul 9–14; Toronto, ON, Canada. p. 9802–22. doi:10.18653/v1/2023.acl-long.546.
23. Desai S, Durrett G. Calibration of pre-trained transformers. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020 Nov 16–20; Online. p. 295–302. doi:10.18653/v1/2020.emnlp-main.21.
24. Breck E, Polyzotis N, Roy S, Whang SE, Zinkevich MA. Data validation for machine learning. In: Proceedings of the 2019 USENIX Conference on Operational Machine Learning; 2019 May 20; Santa Clara, CA, USA.
25. Sambasivan N, Kapania S, Highfill H, Akrong D, Paritosh P, Aroyo LM. "Everyone wants to do the model work, not the data work": data cascades in high-stakes AI. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems; 2021 May 8–13; Yokohama, Japan. p. 1–15. doi:10.1145/3411764.3445518.
26. Zhang JM, Harman M, Ma L, Liu Y. Machine learning testing: survey, landscapes and horizons. IEEE Trans Softw Eng. 2022;48(1):1–36. doi:10.1109/TSE.2019.2962027.
27. Evidently AI Team. Model monitoring for ML in production: a comprehensive guide [Internet]. 2025 [cited 2026 Feb 17]. Available from: https://www.evidentlyai.com/ml-in-production/model-monitoring.
28. Paleyes A, Urma RG, Lawrence ND. Challenges in deploying machine learning: a survey of case studies. ACM Comput Surv. 2023;55(6):1–29. doi:10.1145/3533378.
29. Miao X, Oliaro G, Zhang Z, Cheng X, Jin H, Chen T, et al. Towards efficient generative large language model serving: a survey from algorithms to systems. ACM Comput Surv. 2025;58(1):1–37. doi:10.1145/3754448.
30. Xu Y, Li M, Cui L, Huang S, Wei F, Zhou M. LayoutLM: pre-training of text and layout for document image understanding. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2020 Jul 6–10; San Diego, CA, USA. p. 1192–200. doi:10.1145/3394486.3403172.
31. Masry A, Long DX, Tan JQ, Joty S, Hoque E. ChartQA: a benchmark for question answering about charts with visual and logical reasoning. In: Findings of the Association for Computational Linguistics: ACL 2022; 2022 May 22–27; Dublin, Ireland. p. 2263–79. doi:10.18653/v1/2022.findings-acl.177.
32. Islam N, Islam Z, Noor N. A survey on optical character recognition system. J Inf Commun Technol. 2016;10(2):1–4.
33. Smith R. An overview of the tesseract OCR engine. In: Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007); 2007 Sep 23–26; Curitiba, Brazil. p. 629–33. doi:10.1109/ICDAR.2007.4376991.

34. Karatzas D, Gomez-Bigorda L, Nicolaou A, Ghosh S, Bagdanov A, Iwamura M, et al. ICDAR 2015 competition on robust reading. In: Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR); 2015 Aug 23–26; Tunis, Tunisia. p. 1156–60. doi:10.1109/ICDAR.2015.7333942.
35. Wan Z, He M, Chen H, Bai X, Yao C. TextScanner: reading characters in order for robust scene text recognition. Proc AAAI Conf Artif Intell. 2020;34(7):12120–7. doi:10.1609/aaai.v34i07.6891.
36. Wang D, Raman N, Sibue M, Ma Z, Babkin P, Kaur S, et al. DocLLM: a layout-aware generative language model for multimodal document understanding. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics; 2024 Aug 11–16; Bangkok, Thailand. p. 8529–48. doi:10.18653/v1/2024.acl-long.463.
37. Li M, Lv T, Chen J, Cui L, Lu Y, Florencio D, et al. TrOCR: transformer-based optical character recognition with pre-trained models. Proc AAAI Conf Artif Intell. 2023;37(11):13094–102. doi:10.1609/aaai.v37i11.26538.
38. Inbasekaran A, Gnanasekaran RK, Marciano R. Using transfer learning to contextually optimize optical character recognition (OCR) output and perform new feature extraction on a digitized cultural and historical dataset. In: Proceedings of the 2021 IEEE International Conference on Big Data; 2021 Dec 15–18; Orlando, FL, USA. p. 2224–30. doi:10.1109/bigdata52589.2021.9671586.
39. Bazzo GT, Lorentz GA, Suarez Vargas D, Moreira VP. Assessing the impact of OCR errors in information retrieval. In: Proceedings of the 42nd European Conference on Information Retrieval (ECIR); 2020 Apr 14–17; Lisbon, Portugal. p. 102–9. doi:10.1007/978-3-030-45442-5_13.
40. Long R, Wang W, Xue N, Gao F, Yang Z, Wang Y, et al. Parsing table structures in the wild. In: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada. p. 924–32. doi:10.1109/ICCV48922.2021.00098.
41. Zhang J, Zhang Q, Wang B, Ouyang L, Wen Z, Li Y, et al. OCR hinders RAG: evaluating the cascading impact of OCR on retrieval-augmented generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV); 2025 Oct 19–23; Honolulu, HI, USA. p. 17443–53.
42. Xu Y, Xu Y, Lv T, Cui L, Wei F, Wang G, et al. LayoutLMv2: multi-modal pre-training for visually-rich document understanding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing; 2021 Aug 1–6; Online. p. 2579–91. doi:10.18653/v1/2021.acl-long.201.
43. Huang Y, Lv T, Cui L, Lu Y, Wei F. LayoutLMv3: pre-training for document AI with unified text and image masking. In: Proceedings of the 30th ACM International Conference on Multimedia; 2022 Oct 10–14; Lisboa, Portugal. p. 4083–91. doi:10.1145/3503161.3548112.
44. Appalaraju S, Jasani B, Kota BU, Xie Y, Manmatha R. DocFormer: end-to-end transformer for document understanding. In: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada. p. 973–83. doi:10.1109/ICCV48922.2021.00103.
45. Wei H, Liu C, Chen J, Wang J, Kong L, Xu Y, et al. General OCR theory: towards OCR-2.0 via a unified end-to-end model. arXiv:2409.01704. 2024. doi:10.48550/arXiv.2409.01704.
46. Kim G, Hong T, Yim M, Nam J, Park J, Yim J, et al. OCR-free document understanding transformer. In: Computer Vision—ECCV 2022. Cham, Switzerland: Springer; 2022. p. 498–517. doi:10.1007/978-3-031-19815-1_29.
47. Mathew M, Karatzas D, Jawahar CV. DocVQA: a dataset for VQA on document images. In: Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV); 2021 Jan 5–9; Waikoloa, HI, USA. p. 2199–208. doi:10.1109/wacv48630.2021.00225.
48. Cui L, Xu Y, Lv T, Wei F. Document AI: benchmarks, models and applications. arXiv:2111.08609. 2021. doi:10.48550/arxiv.2111.08609.
49. Wang B, Wu B, Li W, Fang M, Huang Z, Huang J, et al. Infinity parser: layout aware reinforcement learning for scanned document parsing. arXiv:2506.03197. 2025. doi:10.48550/arxiv.2506.03197.
50. Farquhar S, Kossen J, Kuhn L, Gal Y. Detecting hallucinations in large language models using semantic entropy. Nature. 2024;630(8017):625–30. doi:10.1038/s41586-024-07421-0.
51. Li J, Chen J, Ren R, Cheng X, Zhao X, Nie JY, et al. The dawn after the dark: an empirical study on factuality hallucination in large language models. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics; 2024 Aug 11–16; Bangkok, Thailand. p. 10879–99. doi:10.18653/v1/2024.acl-long.586.

52. Ren R, Qu Y, Liu J, Zhao X, Wu Q, Ding Y, et al. A thorough examination on zero-shot dense retrieval. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023; 2023 Dec 6–10; Singapore. p. 15783–96. doi:10.18653/v1/2023.findings-emnlp.1057.
53. Liu YA, Zhang R, Guo J, de Rijke M, Fan Y, Cheng X. Robust neural information retrieval: an adversarial and out-of-distribution perspective. ACM Trans Inf Syst. 2026;44(1):1–48. doi:10.1145/3768153.
54. Robertson S, Zaragoza H. The probabilistic relevance framework: BM25 and beyond. Found Trends Inf Retr. 2009;4(1–2):1–174. doi:10.1561/1500000019.
55. Formal T, Zhan J, Maillard J, Gallé M, Clinchant S. SPLADE: sparse lexical and expansion model for information retrieval. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2021 Jul 11–15; Online. p. 4360–74.
56. Karpukhin V, Oguz B, Min S, Lewis P, Wu L, Edunov S, et al. Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020 Nov 16–20; Online. p. 6769–81. doi:10.18653/v1/2020.emnlp-main.550.
57. Wang S, Zhao Y, Tang D, Xie X, Zhang T. Text embeddings by weakly supervised contrastive pre-training. arXiv:2212.03533. 2022. doi:10.48550/arxiv.2212.03533.
58. Khattab O, Zaharia M. ColBERT: efficient and effective passage search via contextualized late interaction over BERT. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2020 Jul 25–30; Online. p. 39–48. doi:10.1145/3397271.3401075.
59. Cormack GV, Clarke CLA, Buettcher S. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2009 Jul 19–23; Boston, MA, USA. p. 758–9. doi:10.1145/1571941.1572114.
60. Nogueira R, Cho K. Passage re-ranking with BERT. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2019 Jul 21–25; Paris, France. p. 127–36.
61. Nogueira R, Jiang Z, Pradeep R, Lin J. Document ranking with a pretrained sequence-to-sequence model. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020; 2020 Nov 16–20; Online. p. 708–18. doi:10.18653/v1/2020.findings-emnlp.63.
62. Izacard G, Grave E. Leveraging passage retrieval with generative models for open domain question answering. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics; 2021 Apr 19–23; Online. p. 874–80. doi:10.18653/v1/2021.eacl-main.74.
63. Bu C, Chang G, Chen Z, Dang CY, Wu Z, He Y, et al. Query-driven multimodal GraphRAG: dynamic local knowledge graph construction for online reasoning. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025; 2025 Jul 27–Aug 1; Vienna, Austria. p. 21360–80. doi:10.18653/v1/2025.findings-acl.1100.
64. Guo K, Shomer H, Zeng S, Han H, Wang Y, Tang J. Empowering GraphRAG with knowledge filtering and integration. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing; 2025 Nov 4–9; Suzhou, China. p. 25439–53. doi:10.18653/v1/2025.emnlp-main.1293.
65. Liu NF, Lin K, Hewitt J, Paranjape A, Bevilacqua M, Petroni F, et al. Lost in the middle: how language models use long contexts. Trans Assoc Comput Linguist. 2024;12(5):157–73. doi:10.1162/tacl_a_00638.
66. Jin J, Li X, Dong G, Zhang Y, Zhu Y, Wu Y, et al. Hierarchical document refinement for long-context retrieval-augmented generation. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics; 2025 Jul 27–Aug 1; Vienna, Austria. p. 3502–20. doi:10.18653/v1/2025.acl-long.176.
67. Qi P, Lin X, Mehr L, Wang Z, Manning CD. Answering complex open-domain questions through iterative query generation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing; 2019 Nov 3–7. Hong Kong, China. p. 2590–602.
68. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. 2020;21(140):1–67.
69. Thakur N, Reimers N, Rücklé A, Srivastava A, Gurevych I. BEIR: a heterogeneous benchmark for zero-shot evaluation of information retrieval models. In: Proceedings of the Advances in Neural Information Processing Systems; 2021 Dec 6–14; Online.

70. Hofstätter S, Althammer S, Schröder M, Sertkan M, Hanbury A. Improving efficient neural ranking models with cross-architecture knowledge distillation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2021 Jul 11–15; Online.
71. De Cao N, Aziz W, Titov I. Autoregressive entity retrieval. In: Proceedings of the International Conference on Learning Representations (ICLR); 2021 May 3–7; Online.
72. Sun H, Bedrax-Weiss T, Cohen W. PullNet: open domain question answering with iterative retrieval on knowledge bases and text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019 Nov 3–7; Hong Kong, China. p. 2380–90. doi:10.18653/v1/D19-1242.
73. Das R, Neelakantan A, Belanger D, McCallum A. Chains of reasoning over entities, relations, and text using recurrent neural networks. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics; 2017 Apr 3–7; Valencia, Spain. p. 132–41.
74. He X, Tian Y, Sun Y, Chawla NV, Laurent T, LeCun Y, et al. G-retriever: retrieval-augmented generation for textual graph understanding and question answering. Adv Neural Inf Process Syst. 2024;37:132876–907. doi:10.52202/079017-4224.
75. Yoran O, Wolfson T, Ram O, Berant J. Making retrieval-augmented language models robust to irrelevant context. In: Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024); 2024 May 7–11; Vienna, Austria.
76. Kwiatkowski T, Palomaki J, Redfield O, Collins M, Parikh A, Alberti C, et al. Natural questions: a benchmark for question answering research. Trans Assoc Comput Linguist. 2019;7(15):453–66. doi:10.1162/tacl_a_00276.
77. Beltagy I, Peters ME, Cohan A. Longformer: the long-document transformer. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020 Jul 5–10; Online. p. 844–55.
78. Zhuang Z, Zhang Z, Cheng S, Yang F, Liu J, Huang S, et al. EfficientRAG: efficient retriever for multi-hop question answering. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; 2024 Nov 12–16; Miami, FL, USA. p. 3392–411. doi:10.18653/v1/2024.emnlp-main.199.
79. Trivedi H, Balasubramanian N, Khot T, Sabharwal A. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics; 2023 Jul 9–14; Toronto, ON, Canada. p. 10014–37. doi:10.18653/v1/2023.acl-long.557.
80. Press O, Zhang M, Min S, Schmidt L, Smith N, Lewis M. Measuring and narrowing the compositionality gap in language models. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023; 2023 Dec 6–10; Singapore. p. 5687–711. doi:10.18653/v1/2023.findings-emnlp.378.
81. Khandelwal U, He H, Qi P, Jurafsky D. Sharp nearby, fuzzy far away: how neural language models use context. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics; 2018 Jul 15–20; Melbourne, Australia. p. 284–94. doi:10.18653/v1/P18-1027.
82. Press O, Smith NA, Lewis M. Train short, test long: attention with linear biases enables input length extrapolation. In: Proceedings of the International Conference on Learning Representations; 2022 Apr 25–29; Online.
83. MacDonald C, Tonellotto N, Ounis I. Efficient and effective selective query rewriting with efficiency predictions. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2017 Aug 7–11; Tokyo, Japan. p. 495–504. doi:10.1145/3077136.3080827.
84. Zhang Z, Fang M, Chen L. RetrievalQA: assessing adaptive retrieval-augmented generation for short-form open-domain question answering. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024; 2024 Aug 11–16; Bangkok, Thailand. p. 6963–75. doi:10.18653/v1/2024.findings-acl.415.
85. Singh A, Ehtesham A, Kumar S, Khoei TT. Agentic retrieval-augmented generation: a survey on agentic RAG. arXiv:2501.09136. 2025. doi:10.48550/arxiv.2501.09136.
86. Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, et al. ReAct: synergizing reasoning and acting in language models. In: Proceedings of the International Conference on Learning Representations (ICLR); 2023 May 1–5; Kigali, Rwanda.

87. Cancedda N, Dessi R, Dwivedi-Yu J, Hambro E, Lomeli M, Raileanu R, et al. Toolformer: language models can teach themselves to use tools. In: Proceedings of the Advances in Neural Information Processing Systems 36; 2023 Dec 10–16; New Orleans, LA, USA. p. 68539–51. doi:10.52202/075280-2997.
88. LangChain. LangGraph: stateful multi-actor applications with LLMs [Internet]. 2023 [cited 2026 Feb 16]. Available from: https://github.com/langchain-ai/langgraph.
89. Wu Q, Bansal G, Zhang J, Wu Y, Li B, Zhu E, et al. AutoGen: enabling next-gen LLM applications via multi-agent conversation. arXiv:2308.08155. 2023. doi:10.48550/arxiv.2308.08155.
90. Sapkota R, Roumeliotis KI, Karkee M. AI agents vs. agentic AI: a conceptual taxonomy, applications and challenges. Inf Fusion. 2025;104:103599. doi:10.1016/j.inffus.2025.103599.
91. Aquino GA, Ribeiro M, Valente A. From RAG to multi-agent systems: a survey of modern approaches in large language model development. Preprints. 2025. doi:10.20944/preprints202502.0406.v1.
92. Huang W, Abbeel P, Pathak D, Mordatch I. Language models as zero-shot planners: extracting actionable knowledge for embodied agents. In: Proceedings of the 39th International Conference on Machine Learning; 2022 Jul 17–23; Baltimore, MD, USA. p. 9118–47.
93. Valmeekam K, Olmo A, Sreedharan S, Kambhampati S. Large language models still can't plan (a benchmark for LLM planning). In: Proceedings of the NeurIPS, 2023 Workshop on Foundation Models; 2023 Dec 10–16; New Orleans, LA, USA.
94. Rashkin H, Nikolaev V, Lamm M, Aroyo L, Collins M, Das D, et al. Measuring attribution in natural language generation models. Comput Linguist. 2023;49(4):777–840. doi:10.1162/coli_a_00486.
95. Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C, Mishkin P, et al. Training language models to follow instructions with human feedback. Adv Neural Inf Process Syst. 2022;35:27730–44. doi:10.52202/068431-2011.
96. Min S, Krishna K, Lyu X, Lewis M, Yih WT, Koh PW, et al. FActScore: fine-grained atomic evaluation of factual precision in long form text generation. arXiv:2305.14251. 2023. doi:10.48550/arxiv.2305.14251.
97. Lyu Q, Havaldar S, Stein A, Zhang L, Rao D, Wong E, et al. Faithful chain-of-thought reasoning. In: Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics; 2023 Nov 1–4; Nusa Dua, Indonesia. p. 305–29. doi:10.18653/v1/2023.ijcnlp-main.20.
98. Dhuliawala S, Komeili M, Xu J, Raileanu R, Li X, Celikyilmaz A, et al. Chain-of-verification reduces hallucination in large language models. arXiv:2309.11495. 2023. doi:10.48550/arxiv.2309.11495.
99. Gao L, Dai Z, Pasupat P, Chen A, Chaganty AT, Fan Y, et al. RARR: researching and revising what language models say, using language models. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics; 2023 Jul 9–14; Toronto, ON, Canada. p. 16477–508. doi:10.18653/v1/2023.acl-long.910.
100. Liu Y, Iter D, Xu Y, Wang S, Xu R, Zhu C. G-Eval: NLG evaluation using GPT-4 with better human alignment. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing; 2023 Dec 6–10; Singapore. p. 2511–22. doi:10.18653/v1/2023.emnlp-main.153.
101. Yuan W, Pang RY, Cho K, Li X, Sukhbaatar S, Xu J, et al. Self-rewarding language models. In: Proceedings of the 41st International Conference on Machine Learning (ICML'24); 2024 Jul 21–27; Vienna, Austria. p. 57905–23.
102. Reiter E. A structured review of the validity of BLEU. Comput Linguist. 2018;44(3):393–401. doi:10.1162/coli_a_00322.
103. Sellam T, Das D, Parikh A. BLEURT: learning robust metrics for text generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020 Jul 5–10; Online. p. 7881–92. doi:10.18653/v1/2020.acl-main.704.
104. Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y. BERTScore: evaluating text generation with BERT. In: Proceedings of the International Conference on Learning Representations (ICLR); 2020 Apr 26–30; Online.
105. Zheng L, Chiang WL, Sheng Y, Zhuang S, Wu Z, Zhuang Y, et al. Judging LLM-as-a-judge with MT-bench and chatbot arena. Adv Neural Inf Process Syst. 2023;36:46595–623. doi:10.52202/075280-2020.
106. Kiela D, Bartolo M, Nie Y, Kaushik D, Geiger A, Wu Z, et al. Dynabench: rethinking benchmarking in NLP. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational

Linguistics: Human Language Technologies; 2021 Jun 6–11; Online. p. 4110–24. doi:10.18653/v1/2021.naacl-main.324.

107. Amershi S, Begel A, Bird C, DeLine R, Gall H, Kamar E, et al. Software engineering for machine learning: a case study. In: Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP); 2019 May 25–31; Montreal, QC, Canada. p. 291–300. doi:10.1109/ICSE-SEIP.2019.00042.
108. Es S, James J, Espinosa Anke L, Schockaert S. *RAGAs*: automated evaluation of retrieval augmented generation. In: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics; 2024 Mar 17–22; St. Julians, Malta. p. 150–8. doi:10.18653/v1/2024.eacl-demo.16.
109. Manakul P, Liusie A, Gales M. SelfCheckGPT: zero-resource black-box hallucination detection for generative large language models. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing; 2023 Dec 6–10; Singapore. p. 9004–17. doi:10.18653/v1/2023.emnlp-main.557.
110. Mökander J, Schuett J, Kirk HR, Floridi L. Auditing large language models: a three-layered approach. AI Ethics. 2024;4(4):1085–115. doi:10.1007/s43681-023-00289-2.
111. Ru D, Qiu L, Hu X, Zhang T, Shi P, Chang S, et al. RAGChecker: a fine-grained framework for diagnosing retrieval-augmented generation. Adv Neural Inf Process Syst. 2024;37:21999–2027. doi:10.52202/079017-0692.
112. Cassano F, Gopinath A, Narasimhan K, Shinn N, Yao S. Reflexion: language agents with verbal reinforcement learning. In: Advances in Neural Information Processing Systems 36; 2023 Dec 10–16; New Orleans, LA, USA. p. 8634–52. doi:10.52202/075280-0377.
113. Saad-Falcon J, Khattab O, Potts C, Zaharia M. ARES: an automated evaluation framework for retrieval-augmented generation systems. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2024 Jun 16–21; Mexico City, Mexico. p. 338–54. doi:10.18653/v1/2024.naacl-long.20.
114. Qi J, Sarti G, Fernández R, Bisazza A. Model internals-based answer attribution for trustworthy retrieval-augmented generation. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; 2024 Nov 12–16; Miami, FL, USA. p. 6037–53. doi:10.18653/v1/2024.emnlp-main.347.
115. Kim J, Hur M, Min M. From RAG to QA-RAG: integrating generative AI for pharmaceutical regulatory compliance process. In: Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing; 2025 Mar 31–Apr 4; Catania, Italy. p. 1293–5. doi:10.1145/3672608.3707749.
116. Zhao P, Zhang H, Yu Q, Wang Z, Geng Y, Fu F, et al. Retrieval-augmented generation for AI-generated content: a survey. Data Sci Eng. 2026;11(1):1–29. doi:10.1007/s41019-025-00335-5.
117. Niu C, Wu Y, Zhu J, Xu S, Shum K, Zhong R, et al. RAGTruth: a hallucination corpus for developing trustworthy retrieval-augmented language models. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics; 2024 Aug 11–16; Bangkok, Thailand. p. 10862–78.
118. Hendrycks D, Burns C, Chen A, Ball S. CUAD: an expert-annotated NLP dataset for legal contract review. In: Proceedings of the Advances in Neural Information Processing Systems 34 (NeurIPS); 2021 Dec 6–14; Online.
119. Chen Z, Chen W, Smiley C, Shah S, Borova I, Langdon D, et al. FinQA: a dataset of numerical reasoning over financial data. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing; 2021 Nov 7–11; Online. p. 3697–711. doi:10.18653/v1/2021.emnlp-main.300.
120. Guha N, Nyarko J, Ho D, Ré C, Chilton A, Chohlas-Wood A, et al. Legalbench: a collaboratively built benchmark for measuring legal reasoning in large language models. Adv Neural Inf Process Syst. 2023;36:44123–279. doi:10.52202/075280-1915.
121. Yin S, Fu C, Zhao S, Li K, Sun X, Xu T, et al. A survey on multimodal large language models. Natl Sci Rev. 2024;11(12):nwae403. doi:10.1093/nsr/nwae403.
122. Floridi L, Cowls J, Beltrametti M, Chatila R, Chazerand P, Dignum V, et al. AI4People—an ethical framework for a good AI society: opportunities, risks, principles, and recommendations. Minds Mach. 2018;28(4):689–707. doi:10.1007/s11023-018-9482-5.

123. McIntosh TR, Susnjak T, Liu T, Watters P, Xu D, Liu D, et al. From COBIT to ISO 42001: evaluating cybersecurity frameworks for opportunities, risks, and regulatory compliance in commercializing large language models. Comput Secur. 2024;144(8):103964. doi:10.1016/j.cose.2024.103964.
124. Nannini L. Habemus a right to an explanation: so what? A framework on transparency-explainability functionality and tensions in the EU AI act. ACM Conf AI Ethics Soc. 2024;7:1023–35. doi:10.1609/aies.v7i1.31700.
125. Rahwan I, Cebrian M, Obradovich N, Bongard J, Bonnefon JF, Breazeal C, et al. Machine behaviour. Nature. 2019;568(7753):477–86. doi:10.1038/s41586-019-1138-y.
126. Edwards L, Veale M. Slave to the algorithm? Why a 'right to an explanation' is probably not the remedy you are looking for. Duke L Tech Rev. 2017;16(1):18–84.
127. Casper S, Ezell C, Siegmann C, Kolt N, Curtis TL, Bucknall B, et al. Black-box access is insufficient for rigorous AI audits. In: Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency; 2024 Jun 3–6; Rio de Janeiro, Brazil. p. 2254–72. doi:10.1145/3630106.3659037.
128. Wu K, Pang L, Shen H, Cheng X. Enhancing training data attribution for large language models with fitting error consideration. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; 2024 Nov 12–16; Miami, FL, USA. p. 14131–43. doi:10.18653/v1/2024.emnlp-main.782.
129. Lau GKR, Niu X, Dao H, Chen J, Foo CS, Low BKH. Waterfall: scalable framework for robust text watermarking and provenance for LLMs. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; 2024 Nov 12–16; Miami, FL, USA. p. 20432–66. doi:10.18653/v1/2024.emnlp-main.1138.
130. Xiao Y, Jin Y, Bai Y, Wu Y, Yang X, Luo X, et al. Large language models can be contextual privacy protection learners. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; 2024 Nov 12–16; Miami, FL, USA. p. 14179–201. doi:10.18653/v1/2024.emnlp-main.785.
131. Chen R, Hu T, Feng Y, Liu Z. Learnable privacy neurons localization in language models. arXiv:2405.10989. 2024. doi:10.48550/arxiv.2405.10989.
132. Rebedea T, Dinu R, Sreedhar M, Parisien C, Cohen J. *NeMo* guardrails: a toolkit for controllable and safe LLM applications with programmable rails. arXiv:2310.10501. 2023. doi:10.48550/arXiv.2310.10501.
133. Allen Institute for AI. WildGuard: open one-stop moderation tools for safety risks, jailbreaks, and refusals [Internet]. 2024 [cited 2025 Mar 8]. Available from: https://github.com/allenai/wildguard.
134. Shang Z, Ke W, Xiu N, Wang P, Liu J, Li Y, et al. OntoFact: unveiling fantastic fact-skeleton of LLMs via ontology-driven reinforcement learning. Proc AAAI Conf Artif Intell. 2024;38(17):18934–43. doi:10.1609/aaai.v38i17.29859.
135. Mazeika M, Phan L, Yin X, Zou A, Wang Z, Mu N, et al. HarmBench: a standardized evaluation framework for automated red teaming and robust refusal. arXiv:2402.04249. 2024. doi:10.48550/arXiv.2402.04249.
136. Zhang J, Zhou Y, Liu Y, Li Z, Hu S. Holistic automated red teaming for large language models through top-down test case generation and multi-turn interaction. arXiv:2409.16783. 2024. doi:10.48550/arxiv.2409.16783.
137. Suri M, Mathur P, Dernoncourt F, Goswami K, Rossi RA, Manocha D. VisDoM: multi-document QA with visually rich elements using multimodal retrieval-augmented generation. In: Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies; 2025 Apr 29–May 4; Albuquerque, NM, USA. p. 6088–109. doi:10.18653/v1/2025.naacl-long.310.
138. Jin Z, Cao P, Chen Y, Liu K, Jiang X, Xu J, et al. Tug-of-war between knowledge: exploring and resolving knowledge conflicts in retrieval-augmented language models. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024); 2024 May 20–25; Torino, Italy. p. 16867–78.
139. Shin J, Park C, Park J, Seo J, Lim H. MultiDocFusion: hierarchical and multimodal chunking pipeline for enhanced RAG on long industrial documents. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing; 2025 Apr 29–May 4; Albuquerque, NM, USA. p. 20985–1004. doi:10.18653/v1/2025.emnlp-main.1062.
140. Rashkin H, Reitter D, Tomar GS, Das D. Increasing faithfulness in knowledge-grounded dialogue with controllable features. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing; 2021 Aug 1–6; Online. p. 704–18. doi:10.18653/v1/2021.acl-long.58.

141. Liang P, Bommasani R, Lee T, Tsipras D, Soylu D, Yasunaga M, et al. Holistic evaluation of language models. arXiv:2211.09110. 2022. doi:10.48550/arXiv.2211.09110.
142. Thrush T, Jiang R, Bartolo M, Singh A, Williams A, Kiela D, et al. Winoground: probing vision and language models for visio-linguistic compositionality. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. p. 5228–38. doi:10.1109/CVPR52688.2022.00517.
143. Bender EM, Gebru T, McMillan-Major A, Shmitchell S. On the dangers of stochastic parrots: can language models be too big? In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency; 2021 Mar 3–10; Online. p. 610–23. doi:10.1145/3442188.3445922.
144. Yan SQ, Gu JC, Zhu Y, Ling ZH. Corrective retrieval augmented generation. In: Proceedings of the Thirteenth International Conference on Learning Representations (ICLR 2025); 2025 Apr 24–28; Singapore. p. 12845–59.
145. Asai A, Wu Z, Wang Y, Sil A, Hajishirzi H. Self-RAG: learning to retrieve, generate, and critique through self-reflection. arXiv:2310.11511. 2023. doi:10.48550/arXiv.2310.11511.
146. Johnson J, Douze M, Jégou H. Billion-scale similarity search with GPUs. IEEE Trans Big Data. 2021;7(3):535–47. doi:10.1109/TBDATA.2019.2921572.
147. Dao T, Ermon S, Fu D, Ré C, Rudra A. FlashAttention: fast and memory-efficient exact attention with IO-awareness. In: Proceedings of the Advances in Neural Information Processing Systems 35; 2022 Nov 28–Dec 9; New Orleans, LA, USA. p. 16344–59. doi:10.52202/068431-1189.
148. Zhu C, Ping W, Xiao C, Shoeybi M, Goldstein T, Anandkumar A, et al. Long-short transformer: efficient transformers for language and vision. Adv Neural Inf Process Syst. 2021;34:17723–36.
149. Dettmers T, Pagnoni A, Holtzman A, Zettlemoyer L. QLoRA: efficient finetuning of quantized LLMs. Adv Neural Inf Process Syst. 2023;36:10088–115. doi:10.52202/075280-0441.
150. Frantar E, Ashkboos S, Hoefler T, Alistarh D. GPTQ: accurate post-training quantization for generative pre-trained transformers. arXiv:2210.17323. 2022. doi:10.48550/arXiv.2210.17323.
151. Borgeaud S, Mensch A, Hoffmann J, Cai T, Rutherford E, Millican K, et al. Improving language models by retrieving from trillions of tokens. Proc Mach Learn Res. 2022;162:2206–40.
152. Dao T. FlashAttention-2: faster attention with better parallelism and work partitioning. In: Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024); 2024 May 7–11; Vienna, Austria.
153. Chen Y, Qian S, Tang H, Lai X, Liu Z, Han S, et al. LongLoRA: efficient fine-tuning of long-context large language models. arXiv:2309.12307. 2023. doi:10.48550/arxiv.2309.12307.
154. Gao L, Ma X, Lin J, Callan J. Precise zero-shot dense retrieval without relevance labels. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics; 2023 Jul 9–14; Toronto, ON, Canada. p. 1762–77. doi:10.18653/v1/2023.acl-long.99.
155. Peng B, Alcaide E, Anthony Q, Albalak A, Arcadinho S, Biderman S, et al. RWKV: reinventing RNNs for the transformer era. arXiv:2305.13048. 2023. doi:10.48550/arXiv.2305.13048.
156. Fu Y, Peng H, Ouyang T, Xu Y, Li B. Specializing smaller language models towards multi-step reasoning. In: Proceedings of the 40th International Conference on Machine Learning; 2023 Jul 23–29; Honolulu, HI, USA. p. 3653–68.
157. Madaan A, Tandon N, Gupta P, Hallinan S, Gao L, Wiegreffe S, et al. Self-refine: iterative refinement with self-feedback. Adv Neural Inf Process Syst. 2023;36:46534–94. doi:10.52202/075280-2019.
158. Khattab O, Singhvi A, Maheshwari P, Zhang Z, Santhanam K, Vardhamanan S, et al. DSPy: compiling declarative language model calls into self-improving pipelines. arXiv:2310.03714. 2023. doi:10.48550/arXiv.2310.0371.
159. Qin Y, Hu S, Lin Y, Chen W, Ding N, Cui G, et al. Tool learning with foundation models. ACM Comput Surv. 2023;57(4):1–40. doi:10.1145/3704435.
160. Jimenez C, Lieret K, Narasimhan K, Press O, Wettig A, Yang J, et al. SWE-agent: agent-computer interfaces enable automated software engineering. In: Proceedings of the Advances in Neural Information Processing Systems 37; 2024 Dec 10–15; Vancouver, BC, Canada. p. 50528–652. doi:10.52202/079017-1601.

161. Autio C, Kurzon E, Roberts K, Schwartz R, Dunietz J, Jain S, et al. Artificial intelligence risk management framework: generative artificial intelligence profile. Gaithersburg, MD, USA: National Institute of Standards and Technology; 2024. Report No.: NIST AI 600-1.
162. Chen W, Su Y, Zuo J, Yang C, Yuan C, Chan CM, et al. AgentVerse: facilitating multi-agent collaboration and exploring emergent behaviors. arXiv:2308.10848. 2023. doi:10.48550/arxiv.2308.10848.
163. ISO/IEC 42001:2023. Artificial intelligence—management system. Geneva, Switzerland: International Organization for Standardization; 2023.
164. Sun J, Xu C, Tang L, Wang S, Lin C, Gong Y, et al. Think-on-graph: deep and responsible reasoning of large language models on knowledge graphs. In: Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024); 2024 May 7–11; Vienna, Austria.
165. Luo L, Li YF, Haffari G, Pan S. Reasoning on graphs: faithful and interpretable large language model reasoning. In: Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024); 2024 May 7–11; Vienna, Austria.
166. Zhu X, Xie Y, Liu Y, Li Y, Hu W. Knowledge graph-guided retrieval augmented generation. In: Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies; 2025 Apr 29–May 4; Albuquerque, NM, USA. p. 8912–24. doi:10.18653/v1/2025.naacl-long.449.
167. Hogan A, Blomqvist E, Cochez M, D'amato C, De Melo G, Gutierrez C, et al. Knowledge graphs. ACM Comput Surv. 2022;54(4):1–37. doi:10.1145/3447772.
168. Wang Y, Zhang J, Zhao T, Liu Z. Editable memory graphs. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024); 2024 Nov 12–16; Miami, FL, USA.
169. Park JS, O'Brien J, Cai CJ, Morris MR, Liang P, Bernstein MS. Generative agents: interactive simulacra of human behavior. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology; 2023 Oct 29–2023 Nov 1; San Francisco, CA, USA. p. 1–22. doi:10.1145/3586183.3606763.
170. Yao S, Yu D, Zhao J, Shafran I, Griffiths T, Cao Y, et al. Tree of thoughts: deliberate problem solving with large language models. In: Proceedings of the Advances in Neural Information Processing Systems 36; 2023 Dec 10–16; New Orleans, LA, USA. p. 11809–22. doi:10.52202/075280-0517.
171. Agashe S, Fan Y, Reyna A, Wang XE. LLM-coordination: evaluating and analyzing multi-agent coordination abilities in large language models. In: Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2025; 2025 Apr 29–May 4; Albuquerque, NM, USA. p. 8038–57.
172. Jiang J, Zhou K, Zhao WX, Song Y, Zhu C, Zhu H, et al. KG-agent: an efficient autonomous agent framework for complex reasoning over knowledge graph. arXiv:2402.11163. 2024. doi:10.48550/arxiv.2402.11163.
173. Ning Y, Liu H. UrbanKGent: a unified large language model agent framework for urban knowledge graph construction. Adv Neural Inf Process Syst. 2024;37:123127–54.